

Inference in Sparse Graphs with Pairwise Measurements and Side Information

Dylan J. Foster ^{*} Daniel Reichman [†] Karthik Sridharan [‡]

Abstract

We consider the statistical problem of recovering a hidden “ground truth” binary labeling for the vertices of a graph G up to low Hamming error from noisy edge and vertex measurements. We present new algorithms and a sharp finite-sample analysis for this problem on trees and sparse graphs with poor expansion properties such as hypergrids and ring lattices. Our method generalizes and improves over (Globerson et al., 2015), who introduced the problem for two-dimensional grid lattices.

For trees we provide a simple, efficient, algorithm that infers the ground truth with optimal Hamming error and implies recovery results for *all connected graphs*. Here, the presence of side information is critical to obtain a non-trivial recovery rate. We then show how to adapt this algorithm to tree decompositions of edge-subgraphs of certain graph families such as lattices, resulting in optimal recovery error rates that can be obtained in a highly efficient manner.

The thrust of our analysis is to 1) use the tree decomposition along with edge measurements to produce a small class of viable vertex labelings and 2) apply *fast rate* results from statistical learning theory to show that we can infer the ground truth from this class using vertex measurements. We show the power of our method by providing several examples including hypergrids, ring lattices, and the Newman-Watts model for small world graphs. For two-dimensional grids, our results improve over Globerson et al. (2015) by obtaining optimal recovery in the constant-height regime.

1 Introduction

Statistical inference over graphs and networks is a fundamental problem that has received extensive attention in recent years (Fortunato, 2010; Krzakala et al., 2013; Abbe et al., 2014; Hajek et al., 2014). Typically, these inference problems involve noisy observations of discrete labels assigned to edges of a given network, and our goal is to infer a “ground truth” labeling of the vertices (perhaps up to the right sign) that best explains these observations. Such problems occur in a wide range of disciplines including statistical physics, sociology, community detection, average case analysis, and graph partitioning. This inference problem is also related to machine learning tasks involving structured prediction that arise in computer vision, speech recognition and other applications such as natural language processing. Despite the intractability of maximum likelihood estimation, maximum a-posteriori estimation, and marginal inference in network models in the worst case, it has been observed that approximate inference algorithms work surprisingly well in practice (Sontag et al., 2012), and recent work has focused on improving our theoretical understanding of this phenomenon (Globerson et al., 2015).

^{*}Department of Computer Science, Cornell University. Supported in part by the NDSEG fellowship.

[†]Electrical Engineering and Computer Science, University of California, Berkeley

[‡]Department of Computer Science, Cornell University

A key feature of the inference model introduced in (Globerson et al., 2015) is that in addition to observing noisy edge labels, one also observes noisy vertex labels. Our main motivation in this paper is to consider the extent to which the addition of noisy vertex observations improves our prospects for approximate recovery. Specifically, we consider the following model:

Model 1. We receive an undirected graph $G = (V, E)$ whose vertices are labeled according to an unknown *ground truth* $Y \in \{\pm 1\}^V$. We receive noisy edge measurements $X \in \{\pm 1\}^E$, where $X_{uv} = Y_u Y_v$ with probability $1 - p$ and $X_{uv} = -Y_u Y_v$ otherwise. We receive vertex measurements $Z \in \{\pm 1\}^V$, where $Z_u = Y_u$ with probability $1 - q$ and $Z_u = -Y_u$ otherwise. We assume $q > p$ is constant. Our goal is to produce a labeling $\hat{Y} \in \{\pm 1\}^V$ such that with high probability the Hamming error $\sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v\}$ is bounded by $O(f(p)n)$ where $\lim_{p \rightarrow 0} f(p) = 0$.

As a concrete example, consider the problem of trying to recover opinions of individuals in social networks. Suppose that every individual in a social network can hold one of two opinions labeled by -1 or $+1$. We receive a measurement of whether neighbors in the network have the same opinion, but the value of each measurement is flipped with probability p . Likewise we receive estimates of the opinion of each individual, perhaps using a classification model on their profile, but these estimates are corrupted with probability q . The reader should think of pairwise measurements as being fairly accurate while the vertex estimates as being fairly noisy (flip probability q close to $1/2$). [Model 1](#) then translates to producing an estimate of the opinions of users in the social network which predicts the opinion of few users incorrectly.

A first step in studying recovery problems on graphs with noisy vertex observations was taken by [Globerson et al. \(2014, 2015\)](#) who studied [Model 1](#) on square grids. They prove that the statistical complexity of the problem is essentially determined by the number of cuts with cutset of size k . This observation together with a clever use of planar duality enables them to determine the error rate for the square grid. Furthermore, they use the fact that certain quadratic programs are polytime solvable over planar graphs to devise a polynomial algorithm that attains an optimal rate. We comment that as in [Globerson et al. \(2014, 2015\)](#) we focus on finding a labeling of low Hamming error (as opposed to *exact recovery*, where one seeks to find the error probability that with which we recover all labels correctly). [Chen et al. \(2016\)](#) have recently considered exact recovery for edges in this setting for sparse graphs such as grid and rings. They consider the case where there are *multiple* i.i.d observations of edge labels. In contrast, we focus on the case where there is a single (noisy) observation for each edge, on side information, and on partial recovery.¹

Related community detection models such as the well known Stochastic Block Model (SBM) and Censored Block Model (CBM) consider the case where one wishes to detect two communities based on noisy edge observations. Namely, in these models only noisy edges observations are provided and one wishes to recover the correct labeling of vertices up to sign. Block model literature has focused on graphs which have “good” expansion properties such as complete graphs, random graphs, and spectral expanders. By including side information, our model allows for nontrivial recovery rates and efficient algorithms for graphs with “small” separators such as trees, thin grids, and ring lattices. Studying recovery problems in such “non-expanding” graphs is of interest as many graphs arising in applications such as social networks ([Flaxman, 2007](#)) have poor expansion.

The availability of vertex observations changes the statistical nature of the problem in some important ways. For example, for the n -vertex path, it is not difficult ([Globerson et al., 2014](#)) to show that when there are only noisy edge observations any algorithm will fail to find the correct

¹We refer the reader to [Appendix A](#) for further discussion of related models.

labeling (up to sign) of $\Omega(n)$ edges. In contrast, when noisy vertex observations are available, we can obtain a labeling whose expected Hamming error is at most $O(pn)$. That phenomenon that side information with constant noise level enables partial recovery in virtually all sparsity regimes was also observed in Mossel and Xu (2016).

Challenges and Results The key challenge in designing algorithms for Model 1 is understanding statistical performance: Even for graphs such as trees in which the optimal estimator (the marginalized estimator) can be computed efficiently, it is typically unclear what Hamming error rate this estimator obtains. Our approach is to tackle this statistical challenge directly; we obtain efficient algorithms as a corollary.

Our first observation is that the optimal Hamming error for trees is $\tilde{\Theta}(pn)$ provided q is bounded away from $1/2$ ². This is obtained by an efficient message passing algorithm. We then (efficiently) extend our algorithm for trees to more general graphs using a tree decompositions of (edge)-subgraphs. Our main observation is that a bound on the error rate for each component in the tree decomposition can be lifted to a bound on the error rate for the entire graph by *leveraging side information*. Error bounds for components are easily obtained using cut structure.

This approach has the advantage that it applies to non-planar graphs such as high dimensional grids; it is not clear how to apply the machinery of Globerson et al. (2015) to such graphs both because planar duality no longer applies, and the quadratic program they solve no longer admits an efficient solution. Our decomposition-based approach also enables us to obtain optimal error bounds for thin grids which do not have the so-called weak expansion property that is necessary for the analysis in Globerson et al. (2015).

See Section 4 for an extensive discussion of concrete graph families we consider and the error rates we achieve.

1.1 Preliminaries

We work with an undirected graph $G = (V, E)$, with $|V| = n$ and $|E| = m$. For $W \subseteq V$, we let $G(W)$ be the induced subgraph and $E(W)$ be the edge set of the induced subgraph. Let $N(v)$ be the neighborhood of a vertex v . When it is not clear from context we will use $N_G(v)$ to denote neighborhood with respect to a specific graph G . Likewise, for $S \subseteq V$ we use $\delta_G(S)$ to denote its cut-set (edges with one endpoint in S) with respect to G . For a directed graph, we let $\delta_+(v)$ denote the outgoing neighbors and $\delta_-(v)$ denote the incoming neighbors of v . For a subset $W \subseteq V$ we let $N_G(W) = \bigcup_{v \in W} N_G(v)$. We let $\deg(G)$ denote the maximum degree and Δ_{avg} the average degree.

Parameter range We treat $q = 1/2 - \epsilon$ as constant unless otherwise specified. Furthermore, we shall assume throughout that $p \geq w(1/n)$, so the expected number of edge errors is super-constant. We use \tilde{O} to hide $\log(n)$, $\log(1/p)$, and $1/\epsilon$ factors. We use the phrase “with high probability” to refer to events that occur with probability at most $1 - o_n(1)$.

We assume $\deg(G)$ is constant; in the appendix (Theorem 6) we show that if the minimum degree of the graph is $\Omega(\log n)$ there is a trivial strategy that achieves arbitrarily small Hamming error.

²The assumption on q is necessary as when q approaches $1/2$ it is proven in Globerson et al. (2015) that an error of $\Omega(n)$ is unavoidable for certain trees.

2 Inference for Trees

In this section we show how to efficiently and optimally perform inference in [Model 1](#) when the graph G is a tree. In this case note that the expected number of edges (u, v) of the tree with X_{uv} flipped is $p(n - 1)$. In fact, using a simple Chernoff bound we can show that with high probability, at most $2pn + \tilde{O}(1)$ edges are flipped. This implies that for the ground truth Y , $\sum_{(u,v) \in E} \mathbb{1}\{Y_u \neq X_{u,v}Y_v\} \leq 2pn + \tilde{O}(1)$ with high probability over sampling of the edge labels. Hence to estimate ground truth, it is sufficient to search over labelings \hat{Y} that satisfy the inequality

$$\sum_{(u,v) \in E} \mathbb{1}\{\hat{Y}_u \neq X_{u,v}\hat{Y}_v\} \leq 2pn + \tilde{O}(1). \quad (1)$$

We choose the estimator that is most correlated with the vertex observations Z subject to the aforementioned inequality. That is, we find

$$\hat{Y} = \arg \min_{\hat{Y} \in \{\pm 1\}^V} \sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Z_v\} \quad \text{s.t.} \quad \sum_{(u,v) \in E} \mathbb{1}\{\hat{Y}_u \neq X_{u,v}\hat{Y}_v\} \leq 2pn + \tilde{O}(1) \quad (2)$$

This optimization problem can be solved efficiently — $O(\lceil pn \rceil n^2 \deg(G))$ time for general trees and $O(\lceil pn \rceil n)$ time for stars and line graphs — with message passing. The full algorithm is stated in [Appendix D](#).

On the statistical side we use results from statistical learning theory to show that the Hamming error of \hat{Y} obtained above is with high probability bounded by $\tilde{O}(pn)$. To move to the statistical learning setting (see [Appendix C](#) for an overview) we first define our “hypothesis class” $\mathcal{F} \triangleq \{Y' \in \{\pm 1\}^V \mid \sum_{(u,v) \in E} \mathbb{1}\{Y'_u \neq X_{u,v}Y'_v\} \leq 2pn + \tilde{O}(1)\}$; note that this is precisely the set of Y' satisfying (1). The critical observation here is that for any \hat{Y} the Hamming error (with respect to the ground truth) is proportional to the *excess risk* in the statistical learning setting over Z with class \mathcal{F} :

$$\sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v\} = \frac{1}{1 - 2q} \left[\sum_{v \in V} \mathbb{P}_Z\{\hat{Y}_v \neq Z_v\} - \min_{Y' \in \mathcal{F}} \sum_{v \in V} \mathbb{P}_Z\{Y'_v \neq Z_v\} \right]. \quad (3)$$

Combing (3) with a so-called *fast rate* from statistical learning theory ([Corollary 2](#)) implies that if we take \hat{Y} to be the *empirical risk minimizer* over \mathcal{F} given Z , which is in fact the solution to (2), then we have $\sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v\} \leq O(\log(|\mathcal{F}|/\delta)/\epsilon^2)$ with probability at least $1 - \delta$. Connectivity of G implies $|\mathcal{F}| \approx (\frac{\epsilon}{p})^{2pn + \tilde{O}(1)}$, giving the final $\tilde{O}(pn)$ rate. [Theorem 1](#) makes this result precise:

Theorem 1 (Decoding in Trees). *Let \hat{Y} be the solution to (2). Then with high probability,*

$$\sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v\} \leq \tilde{O}(pn). \quad (4)$$

More precisely, with probability at least $1 - \delta$,

$$\sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v\} \leq \frac{1}{\epsilon^2} (2pn + 2 \log(2/\delta) + 1) \log(2e/p\delta). \quad (5)$$

We remark that side information is critical in this result. For trees — in particular the line graph — no estimator can achieve below $\Omega(n)$ hamming error until $p = O(1/n)$.

3 Inference for General Graphs

3.1 Upper Bound: Inference with Tree Decompositions

Our main algorithm, TREEDECOMPOSITIONDECODER (Algorithm 1) produces estimators for Model 1 for graphs G that admit a *tree decomposition* in the sense of Robertson and Seymour (Robertson and Seymour (1986)). Recall that a tree decomposition for a graph $G = (V, E)$ is new graph $T = (\mathcal{W}, F)$ in which each node in \mathcal{W} corresponds to a subset of nodes in the original graph G . The edge set F forms a tree over \mathcal{W} and must satisfy a property known as *coherence*, which guarantees that the connectivity structure of T captures that of G . The approach of TREEDECOMPOSITIONDECODER is to use the edge observations X to produce a *local* estimator for each component of the tree decomposition T , then use the vertex observations Z to combine the many local estimators into a single *global* estimator.

Tree decompositions have found extensive use in algorithm design and machine learning primarily for computational reasons: These objects allow one to lift algorithmic techniques that are only feasible computationally on constant-sized graphs, such as brute force enumeration, into algorithms that run efficiently on graphs of all sizes. It is interesting to note that our algorithm obeys this principle, but for *statistical performance* in addition to computational performance: We are able to lift an analysis technique that is only tight for constant-sized graphs, the union bound, into an analysis that is tight for arbitrarily large graphs from families such as grids. However, as our analysis for trees shows, this approach is only made possible by the side information Z .

The *width* $\text{wid}(T)$ of a tree decomposition T is the size of the largest component in T , minus one (by convention). To place a guarantee on the performance of TREEDECOMPOSITIONDECODER, both statistically and computationally, it is critical that the width be at most logarithmic in n . At first glance this condition may seem restrictive there are graphs of interests such as grids for which the *treewidth* $\text{tw}(G)$ — the smallest treewidth of *any* tree decomposition — is order \sqrt{n} . For such graphs, our approach is to choose a subset $E' \subseteq E$ of edges to probe so that the graph $G' = (V, E')$ has small treewidth. For all of the graphs we consider this approach obtains optimal sample complexity in spite of discarding information. This is remarkable because for grids this entails discarding a *constant fraction* of the pairwise measurements.

Having found a decomposition of small treewidth for G' we apply the following algorithm. For each component of this decomposition, we compute the maximum likelihood estimator for the labels in this component given the edge measurements X . This is done by brute-force enumeration over vertex labels, which can be done efficiently because we require small treewidth. With these estimates we have a tree decomposition in which for each component, up to sign, we have the right labels for the vertices in the component (to within a tolerable failure probability). Furthermore by concentration of measure we will have that the total number of failures across all components will not be much larger than the expected number of failures.

For a given component, there will be two estimators that match the edges in that component equally well due to sign ambiguity. The remaining problem is to select a set of signs — one for each component — so that local estimators agree globally. For this task we leverage the side information Z . Our approach will mirror that of Section 2: To produce a global prediction \hat{Y} we solve a global optimization problem over the tree decomposition using dynamic programming, then analyze the statistical performance of \hat{Y} using statistical learning theory.

Informally, if there is some Δ such that we can show a p^Δ failure probability for estimating up to sign the vertex labels within each component of the tree decomposition, the prediction produced by Algorithm 1 will attain a high probability $p^\Delta n$ Hamming error bound for the entire graph. For example, in Section 4 we show a p^2 failure probability for estimating vertex labels in a grid of

size 3×2 , which through [Algorithm 1](#) translates to a $O(p^2n)$ rate with high probability on both $\sqrt{n} \times \sqrt{n}$ and $3 \times n/3$ grids.

Definition 1 (E.g., [Cowell et al. \(2006\)](#)). A tree $T = (\mathcal{W}, F)$ is a tree decomposition for $G = (V, E)$ if it satisfies

1. **Vertex Inclusion:** Each node in $v \in V$ belongs to at least one component $W \in \mathcal{W}$.
2. **Edge Inclusion:** For each edge $(u, v) \in E$, there is some $W \in \mathcal{W}$ containing both u and v .
3. **Coherence:** Let $W_1, W_2, W_3 \in \mathcal{W}$ with W_2 on the path between W_1 and W_3 in T . Then if $v \in V$ belongs to W_1 and W_3 , it also belongs to W_2 .

We assume the following additional properties without loss of generality:

- T is not redundant, i.e. there is no $(W, W') \in F$ with $W' \subseteq W$.

The next definition concerns the subsets of the graph G used in the local inference procedure within [Algorithm 1](#). We allow the local maximum likelihood estimator for a component W to consider a superset of nodes, $\text{EXTEND}(W)$, whose definition will be specialized to different classes of graphs.

Definition 2 (Component Extension Function). For a given $W \in \mathcal{W}$, the extended component $W^* \supseteq W$ denotes the result of $\text{EXTEND}(W)$.

Useful choices for the extension function include $\text{EXTEND}(W) = W$ and the neighborhood of W with respect to the probed graph:

$$\text{EXTEND}(W) = W \cup \bigcup_{v \in W} N_{G'}(v). \quad (6)$$

Concrete instantiations of EXTEND are given in [Section 4](#).

We now define quantitative properties of the tree decomposition in [Table 1](#). For a given property, the corresponding (\star) version will denote the analogue that arises in analyzing performance when using extended components. For simplicity, the reader may wish to imagine each (\star) property as the corresponding non- (\star) property on their first read-through.

Table 1: Tree decomposition properties.

$\deg(T) = \max_{W \in \mathcal{W}} \{(W, W') \in F\} $	$\text{wid}^*(T) = \max_{W \in \mathcal{W}} W^* - 1$
$\text{wid}(T) = \max_{W \in \mathcal{W}} W - 1$	$\mathcal{W}^*(e) = \{W \in \mathcal{W} \mid e \in E(W^*)\}$
$\mathcal{W}(e) = \{W \in \mathcal{W} \mid e \in E(W)\}$	$\deg_E^*(T) = \max_{e \in E} \mathcal{W}^*(e) $
$\deg_E(T) = \max_{e \in E} \mathcal{W}(e) $	$\text{mincut}^*(W) = \min_{S \subset W^*, S \cap W \neq \emptyset, \bar{S} \cap W \neq \emptyset} \delta_{G(W)}(S) $
$\text{mincut}(W) = \min_{S \subset W, S \neq \emptyset} \delta_{G(W)}(S) $	

Definition 3 (Admissible Tree Decomposition). We will call a tree decomposition $T = (\mathcal{W}, F)$ admissible if it satisfies the following properties:

- $\deg(T)$, $\deg_E^*(T)$, $\max_{W \in \mathcal{W}} |E(W^*)|$, and $\text{wid}^*(T)$ are constant.
- $G'(W^*)$ is connected for all $W \in \mathcal{W}$ ³.

³Together with our other assumptions, this implies the *connected treewidth* of G' ([Diestel and Müller, 2016](#)) is constant.

In the rest of this section, the \tilde{O} notation will hide all of the constant quantities from [Definition 3](#).

Theorem 2 (Main Theorem). *Let \hat{Y} be the labeling produced using [Algorithm 1](#) with an admissible tree decomposition. Then, with high probability over the draw of X and Z ,*

$$\sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v\} \leq \tilde{O}\left(\sum_{W \in \mathcal{W}} p^{\lceil \text{mincut}^*(W)/2 \rceil}\right). \quad (7)$$

In particular, let Δ be such that $\Delta \leq \text{mincut}^(W)$ for all $W \in \mathcal{W}$. Then, with high probability,*

$$\sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v\} \leq \tilde{O}\left(p^{\lceil \Delta/2 \rceil} n\right). \quad (8)$$

Algorithm 1 runs in time $\tilde{O}(\lceil p^{\Delta/2} n \rceil n^2)$ for general tree decompositions and time $\tilde{O}(\lceil p^{\Delta/2} n \rceil n)$ when T is a line graph.

Algorithm 1 TREEDECOMPOSITIONDECODER

Parameters: Graph $G = (V, E)$. Probed edges $E' \subseteq E$. Extension function EXTEND.

Tree decomposition $T = (\mathcal{W}, F)$ for (V, E') . Failure probability $\delta > 0$.

Input: Edge measurements $X \in \{\pm 1\}^E$. Vertex measurements $Z \in \{\pm 1\}^V$.

```

1: procedure TREEDECOMPOSITIONDECODER
2:   STAGE ONE:
3:   for  $W \in \mathcal{W}$  do                                      $\triangleright$  Solve edge MLE for each component.
4:      $W^* \leftarrow \text{EXTEND}(W)$ .                                $\triangleright$  See Definition 2
5:      $\tilde{Y}^{W^*} \leftarrow \arg \min_{\tilde{Y} \in \{\pm 1\}^{W^*}} \sum_{uv \in E'(W^*)} \mathbb{1}\{\tilde{Y}_u \tilde{Y}_v \neq X_{uv}\}$ .
6:     Let  $\hat{Y}^{W^*}$  be the restriction of  $\tilde{Y}^{W^*}$  to  $W$ .
7:   end for
8:   STAGE TWO:
9:   for  $W \in \mathcal{W}$  do                                      $\triangleright$  Compute array of meta-vertex costs for tree decomposition.
10:     $\text{Cost}_W[+1] \leftarrow \sum_{v \in W} \mathbb{1}\{\hat{Y}_v^{W^*} \neq Z_v\}$    &    $\text{Cost}_W[-1] \leftarrow \sum_{v \in W} \mathbb{1}\{-\hat{Y}_v^{W^*} \neq Z_v\}$ .
11:  end for
12:  for  $(W_1, W_2) \in F$  do                                $\triangleright$  Compute meta-edge costs for tree decomposition.
13:    Let  $v \in W_1 \cap W_2$ .
14:     $S(W_1, W_2) \leftarrow \hat{Y}_v^{W_1^*} \cdot \hat{Y}_v^{W_2^*}$ .
15:  end for
16:   $K_n \leftarrow \lceil \deg(T) (2^{\text{wid}^*(T)+2} \sum_{W \in \mathcal{W}} p^{\lceil \text{mincut}^*(W)/2 \rceil} + 6 \deg_E^*(T) \max_{W \in \mathcal{W}} |E(W^*)| \log(2/\delta)) \rceil$ .
17:   $\hat{s} \leftarrow \text{TREEDECODER}(T, \text{Cost}, S, K_n)$ .    $\triangleright$  Compute optimal signing of components. See Appendix D.
18:  for  $v \in V$  do                                        $\triangleright$  Collapse tree decomposition into final estimator.
19:    Choose an arbitrary  $W$  such that  $v \in W$  and set  $\hat{Y}_v \leftarrow \hat{s}_W \hat{Y}_v^{W^*}$ .
20:  end for
21: return  $\hat{Y}$ .
22: end procedure

```

3.2 Main theorem: Proof sketch

Let us sketch the analysis of [Theorem 2](#) in the simplest case, where $\text{EXTEND}(W) = W$ for all $W \in \mathcal{W}$ and consequently all (\star) properties are replaced with their non- (\star) counterparts. We give a bound begin by bounding that probability that a single component-wise estimator \hat{Y}^W computed on [Line 6](#) of [Algorithm 1](#) fails to exactly recover the ground truth within its component.

Definition 4 (Component Estimator). *The (edge) maximum likelihood estimator for W is given by*

$$\hat{Y}^W \triangleq \arg \min_{\hat{Y} \in \{\pm 1\}^W} \sum_{uv \in E'(W)} \mathbb{1}\{\hat{Y}_u \hat{Y}_v \neq X_{uv}\}. \quad (9)$$

\hat{Y}^W can be computed by enumeration over all labelings in time $2^{|W|}$. There are always two solutions to (9) due to sign ambiguity; we take one arbitrarily.

Proposition 1 (Error Probability for Component Estimator).

$$\mathbb{P} \left(\min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^W \neq Y^W\} > 0 \right) \leq \tilde{O}(p^{\lceil \text{mincut}(W)/2 \rceil})$$

Proof. Assume that both \hat{Y}^W and $-\hat{Y}^W$ disagree with the ground truth or else we are done. Let S be a maximal connected component of the set of vertices v for which $\hat{Y}_v^W \neq Y_v$. It must be the case that at least $\lceil |\delta(S)|/2 \rceil$ edges (u, v) in $\delta(S)$ have X_{uv} flipped from the ground truth, or else we could flip all the vertices in S to get a new estimator that agrees with X better than \hat{Y} ; this would be a contradiction since \hat{Y} minimizes $\sum_{uv \in E'(W)} \mathbb{1}\{\hat{Y}_u \hat{Y}_v \neq X_{uv}\}$. We now apply union bound to get the result:

$$\mathbb{P} \left(\min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^W \neq Y^W\} > 0 \right) \leq \sum_{S \subseteq W: S \neq \emptyset, S \neq W} p^{\lceil |\delta(S)|/2 \rceil} \leq 2^{|W|} p^{\lceil \text{mincut}(W)/2 \rceil}. \quad (10)$$

□

Proposition 1 bounds the probability of failure for *individual components*, but does not immediately imply a bound on the total number of components that may fail for a given realization of X . If the components \mathcal{W} did not overlap one could apply a Chernoff bound to establish such a result, as their predictions would be independent. Since components can in fact overlap their predictions are dependent, but using a more sophisticated concentration inequality (the *entropy method* ([Boucheron et al., 2003](#))) we can show that – so long as no edge appears in too many components – an analogous concentration result holds and total number of components failures is close to the expected number with high probability.

Lemma 1 (Informal). With high probability over the draw of X ,

$$\min_{s \in \{\pm 1\}^{\mathcal{W}}} \sum_{W \in \mathcal{W}} \mathbb{1}\{s_W \hat{Y}^W \neq Y^W\} \leq \tilde{O} \left(\sum_{W \in \mathcal{W}} p^{\lceil \text{mincut}(W)/2 \rceil} \right) \quad (11)$$

In light of (11), consider the signing of the component-wise predictions (\hat{Y}^W) that best matches the ground truth.

$$s^* = \arg \min_{s \in \{\pm 1\}^{\mathcal{W}}} \sum_{W \in \mathcal{W}} \mathbb{1}\{s_W \hat{Y}^W \neq Y^W\}.$$

If we knew the value of s^* we could use it to produce a vertex prediction with a Hamming error bound matching (7). Computing the s^* is information-theoretically impossible because we do not have access to Y . We get the stated result by proceeding in a manner similar to the algorithm (2) for the tree. We first define a class $\mathcal{F} \subseteq \{\pm 1\}^{\mathcal{W}}$ which has the property that 1) $s^* \in \mathcal{F}$ with high probability and 2) $|\mathcal{F}| \lesssim 2^{\tilde{O}(\sum_{W \in \mathcal{W}} 2^{|W|} p^{\lceil \text{mincut}(W)/2 \rceil})}$. Then we take the component labeling \hat{s} computed in Line 17 of Algorithm 1 is simply the element of \mathcal{F} that is most correlated with the vertex observations Z : $\hat{s} = \arg \min_{s \in \mathcal{F}} \sum_{W \in \mathcal{W}} \sum_{v \in W} \mathbb{1}\{s_W \hat{Y}_v^W \neq Z_v\}$. Finally, to produce the final prediction \hat{Y}_v for a given vertex v , we find $W \in \mathcal{W}$ with $v \in W$ and take $\hat{Y}_v = \hat{s}_W \cdot \hat{Y}_v^W$. A generalization bound from statistical learning theory then implies that this predictor enjoys error at most $\tilde{O}(\log |\mathcal{F}|) = \tilde{O}(\sum_{W \in \mathcal{W}} 2^{|W|} p^{\lceil \text{mincut}(W)/2 \rceil})$, which establishes the main theorem.

Efficient implementation Both the tree algorithm and Algorithm 1 rely on solving a constrained optimization problem of the form (2). In Appendix D we show how to perform this procedure efficiently using a message passing scheme.

3.3 Lower Bounds: General Tools

In this section we state simple lower bound techniques for Model 1. Recall that we consider q as a constant, and thus we are satisfied with lower bounds that coincide with our upper bounds up to polynomial dependence on q .

Theorem 3. *Assume $p < q$. Then any algorithm for Model 1 incurs expected hamming error $\Omega(\sum_{v \in V} p^{\lceil \deg(v)/2 \rceil})$.*

Corollary 1. Any algorithm for Model 1 incurs expected hamming error $\Omega(p^{\Delta_{\text{avg}}/2+1} n)$.

Theorem 4. *Let \mathcal{W} be a collection of disjoint constant-sized subsets of V . Then for all p below some constant, any algorithm for Model 1 incurs expected Hamming error $\Omega(\sum_{W \in \mathcal{W}} p^{\lceil |\delta_G(W)|/2 \rceil})$*

4 Concrete Results for Specific Graphs

We now provide tight upper and lower bounds on recovery for concrete classes of graphs.

4.1 Connected Graphs

Example 1 (Arbitrary graphs). *For any connected graph G , the following procedure attains an error rate of $\tilde{O}(pn)$ with high probability:*

1. Find a spanning tree T for G .
2. Run the algorithm from Section 2 on T .

This rate is sharp, in the sense that there are connected graphs — in particular, all trees — for which $\Omega(pn)$ Hamming error is optimal. Furthermore, for all graphs one can attain an estimator whose Hamming error is bounded as $\tilde{O}(pn + \#\text{connected components})$ by taking a spanning tree for each component. This bound is also sharp.

The next example shows that there are connected graphs beyond trees for which $\Omega(pn)$ Hamming error is unavoidable.

Example 2 (Supercritical Erdős-Rényi). Consider the supercritical Erdős-Rényi model $G(n, p)$ where $p = c/n$ and $c > 1$ is a fixed constant. It is well known that with high probability $G(n, p)$ contains a giant component with $\Omega(n)$ vertices, which we call $H(n, p)$. It is known that with high probability $H(n, p)$ contains $\Omega(n)$ vertices of degree at most 2 (Ding et al., 2014; Pittel and Wormald, 2005). It follows that the optimal hamming error for $H(n, p)$ is $\Omega(pn)$. Such error can be attained efficiently by applying the tree algorithm for a spanning tree of $H(n, p)$.

Looking at Theorem 3, one might be tempted to guess that the correct rate for inference is determined entirely by the degree profile of a graph. This would imply, for instance, that for any d -regular graph the correct rate is $\Theta(p^{\lceil d/2 \rceil} n)$. The next example — via Theorem 4 — shows that this is not the case.

Example 3. For any constant d , there exists a family of d -regular graphs on n vertices for which no algorithm in Model 1 attains lower than $\Omega(pn)$ Hamming error.

This construction for $d = 3$ is illustrated in Figure 1. We note that this lower bound is exponentially weak in d , but for constant q and d it is $\Omega(pn)$ nonetheless.

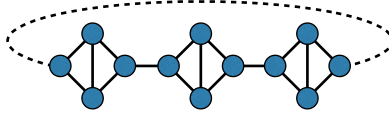


Figure 1: 3-regular graph for which $O(pn)$ error rate is optimal.

4.2 Grid Lattices

In this section we illustrate how to use the tree-decomposition based algorithm, Algorithm 1, to obtain optimal rates for grid lattices.

Example 4 (2-dimensional grid). Let G be a 2-dimensional grid lattice of size $c \times n/c$ where $c \leq \sqrt{n}$. For grid of height $c = 3$ (or above) using Algorithm 1, we obtain an estimator \hat{Y} such that with high probability, the Hamming error is bounded as $O(p^2n)$. This estimator runs in time $O(\lceil p^2n \rceil n)$. By the degree profile argument (also given in Globerson et al. (2015)), there is a matching lower bound of $\Omega(p^2n)$. For a grid of height $c = 1$ there is an obvious lower bound of $\Omega(pn)$ since this graph is a tree.

The estimator of Globerson et al. (2015) can be shown to have expected Hamming error of $O(p^2n)$ for the 2-dimensional grid with $c = \Omega(\log n)$. However, this error rates is only achieved when c order $\log n$ or above. Our method works for constant height grids and with high probability.

Algorithm 1 of course requires a tree decomposition as input. The tree decomposition used to obtain Example 4 for constant-height grids is illustrated in Figure 2 for $c = 3$: The grid is covered in overlapping 3×2 components, and these are connected as a path graph to form the tree decomposition.

The savvy reader will observe that this tree decomposition has $\text{mincut}(W) = 2$, and so only implies a $O(pn)$ Hamming error bound through Theorem 2. This rate falls short of the $O(p^2n)$ rate promised in the example; it is no better than the rate if G were a tree. The problem is that within each 3×2 block, there are four “corner” nodes each with degree 2. Indeed if either edge connected to a corner is flipped from the ground truth, which happens with probability p , this corner is effectively disconnected from the rest of W in terms of information. To sidestep this

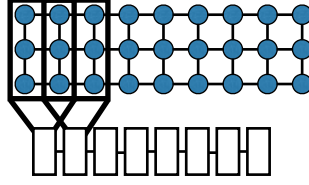


Figure 2: Tree decomposition for $3 \times n/3$ grid

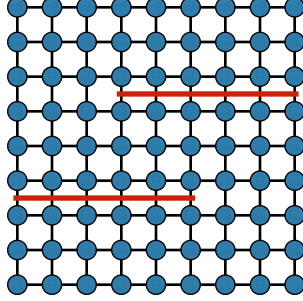


Figure 3: E' for $\sqrt{n} \times \sqrt{n}$ grid.

issue, we define $\text{EXTEND}(W) = \bigcup_{v \in W} N(v)$. With this extension, we have $\text{mincut}^*(W) = 3$ for all components except the endpoints, which implies the $O(p^2n)$ rate.

Probing Edges We now illustrate how to extend the tree decomposition construction for constant-height grids to a construction for grids of arbitrary height. Recall that [Algorithm 1](#) takes as input a subset $E' \subseteq E$ and a tree decomposition T for $G' = (V, E')$. To see where using only a subset of edges can be helpful consider [Figure 2](#) and [Figure 3](#). The $3 \times n/3$ grid is ideal for our decoding approach because it can be covered in 3×2 blocks as in [Figure 2](#) and thus has treewidth at most 5. The $\sqrt{n} \times \sqrt{n}$ grid is more troublesome because it has treewidth \sqrt{n} , but we can arrive at G' with constant treewidth by removing $\Theta(n)$ edges through the “zig-zagging” cut shown in [Figure 3](#). Observe that once the marked edges in [Figure 3](#) are removed we can “unroll” the graph and apply a decomposition similar to [Figure 2](#).

The tree decomposition construction we have outlined for two-dimensional grids readily lifts to higher dimension. This gives rise to the next example.

Example 5 (Hypergrids and Hypertubes). *Consider a three-dimensional grid lattice of of length n/c^2 , height c , and width c . If $c = n^{1/3}$ — that is, we have a cube — then [Algorithm 1](#) obtains Hamming error $\tilde{O}(p^3n)$ with high probability, which is optimal by [Theorem 3](#).*

When c is constant, however, the optimal rate is $\Omega(p^2n)$; this is also obtained by [Algorithm 1](#). This contrasts the two-dimensional grid, where the optimal rate is the same for all $3 \leq c \leq \sqrt{n}$.

[Algorithm 1](#) can be applied to any d -dimensional hypergrid of shape $c \times c \times \dots \times n/c^{d-1}$ to achieve $O(p^d n)$ Hamming error when $c \approx n^{1/d}$. For c constant, the optimal rate is $\Theta(p^{\lceil \frac{d+1}{2} \rceil} n)$. More generally, the optimal rate interpolates between these extremes.

The next two examples briefly sketch how to apply tree decompositions to more lattices. Recall that the triangular lattice and hexagonal lattice are graphs whose drawings can be embedded in \mathbb{R}^2 to form regular triangular and hexagonal tilings, respectively.

Example 6 (Triangular Lattice). Consider a triangular lattice of height and width \sqrt{n} (i.e., a parallelogram with side length \sqrt{n}). Let each component to be a vertex and its 6 neighbors (except for the edges of the mesh), and choose these components such that the graph is covered completely. For a given component, let W^* to be the neighborhood W . One can see that for this decomposition $\text{mincut}^*(W)$ is 6. Hence, [Algorithm 1](#) achieves Hamming error $\tilde{O}(p^3n)$. This rate is optimal because all vertices in the graph have degree 6 besides those at the boundary, but the number of vertices on the boundary is sub-constant.

Example 7 (Hexagonal Lattice). Consider a $\sqrt{n} \times \sqrt{n}$ hexagonal lattice. Take each component W to be a node v and its neighbors, and choose the nodes v so that the graph is covered. Choose W^* to be the neighborhood of the component W . Thus $\text{mincut}^*(W)$ for each component is 3, leading to a Hamming error rate of $\tilde{O}(p^2n)$. This rate is optimal because all vertices on the lattice except those at the boundary have degree 3.

4.3 Newman-Watts Model

To define the Newman-Watts small world model ([Newman and Watts, 1999](#)), we first define the regular ring lattice, which serves as the base graph for this model. The *regular ring lattice* $R_{n,k}$ is a $2k$ -regular graph on n vertices defined as follows: 1) $V = \{1, \dots, n\}$. 2) $E = \{(i, j) \mid j \in \{i+1, \dots, i+k \pmod{n}\}\}$. [Theorem 3](#) immediately implies that the best rate possible in this model is $\Omega(p^k n)$. Using [Algorithm 1](#) with an appropriate decomposition it is indeed possible to achieve this rate.

Example 8. The optimal Hamming rate for $R_{n,k}$ in [Model 1](#) is $\tilde{\Theta}(p^k n)$. Moreover, this rate is achieved by an efficiently by [Algorithm 1](#) in time $O(\lceil p^k n \rceil n)$.

We can now move on to the Newman-Watts model itself:

Definition 5 (Newman-Watts Model). To produce a sample from the Newman-Watts model $H_{n,k,\alpha}$, begin with $R_{n,k}$, then independently replace every non-edge with an edge with probability α/n .

For any constant $\alpha < 1$, a constant fraction of the vertices in $R_{n,k}$ will be untouched in $H_{n,k,\alpha}$. Thus, the inference lower bound for [Example 8](#) still applies, meaning that the optimal rate is $O(p^k n)$. Algorithmically, this result can be obtained by discarding the new edges and using the same decomposition as in [Example 8](#).

Example 9. For any $\alpha < 1$, the optimal Hamming rate for $H_{n,k,\alpha}$ in [Model 1](#) is $\tilde{\Theta}(p^k n)$. Moreover, this rate is achieved by an efficiently by [Algorithm 1](#) in time $O(\lceil p^k n \rceil n)$.

5 Extensions

Semi-random model Our tree decomposition based approach readily extends to the following semi-random model: Each edge is marked independently with probability p . For unmarked edges $(u, v) \in E$ we are provided with the ground truth labeling $Y_u \cdot Y_v$. For marked edges, an adversary is allowed to arbitrarily set labels. For the noisy vertex labels we assume the same random model as in [Model 1](#).

All our results (e.g. [Theorem 2](#)) immediately extend to this setting with the same rates. To see this, observe that if [Lemma 2](#) and [Lemma 4](#) hold in the semi-random setting, then the rest of the analysis can be used unchanged. Now [Lemma 2](#) holds for the semi-random setting for any arbitrary labeling of marked nodes within each component and its extension. This is because the lemma only

depends on bounding probability that more than half the edges in a cut are flipped, and does not depend on *how* marked edges are relabeled. To prove the [Lemma 4](#) in the semi-random setting, we move to an upper bound by allowing edges that appear in multiple components to be labeled differently across components (we can have inconsistent labelings). That is, for each component we consider adversarial labelings local to that component and then use the same concentration result used to prove the bound in [Lemma 4](#).

This observation shows that side information enables tight recovery in the semi-random setting. It should be noted that more common inference methods for the pure edge recovery setting such as spectral algorithms often are not robust to semi-random measurements.

General Sub-Algorithms While [Algorithm 1](#) considers components of constant size and we use a brute-force algorithm to make inference in each component using edge labels, a careful look at our proof technique reveals that one can replace inference within each component by any arbitrary sub-algorithm (e.g. spectral) that obtains exact recovery within each component up to sign (with bounded failure probability). Thus, our decomposition methodology can be used at a high level to combine different inference techniques (based on edge labels) within each components. This approach can apply even when components are allowed to have super-constant size, and suggests an interesting avenue for exploring computational-statistical tradeoffs. We remark that even for the brute force enumeration scheme we use in each component, though we currently assume constant size components, we can easily still produce polynomial time algorithms if the component sizes are allowed to be of order $\log n$.

6 Discussion

We considered [Model 1](#), introduced in [Globerson et al. \(2015\)](#), of approximately inferring the ground truth labels for nodes of a graph based on noisy edge and vertex labels. We make several improvements over [Globerson et al. \(2015\)](#). First, their work provide the right rates for grid graphs and provide upper bounds on recovery rates for planar graphs and expanders. Further, the proof techniques in [Globerson et al. \(2015\)](#) critically relied on a weak expansion property. In contrast, we provide a general method to deal with arbitrary graphs that admit tree decompositions of (edge)-subgraphs. We recover the results in [Globerson et al. \(2015\)](#) for grids, but are able to provide rates for graphs that do not satisfy the weak expansion property. While [Globerson et al. \(2015\)](#) use vertex labels only to break ties (this is unnecessary if wish to recover only up to a global sign), we show that for more general graphs the vertex labels, though very noisy, play a critical role.

Finding Decompositions Throughout this paper we assumed that one could find tree decomposition for the given graph (after deleting some edges) that are admissible in the sense of having low tree width. For all the examples we consider, we directly constructed such decompositions. However, one might ask is whether such a decomposition can be automatically found. That is, in general is there an efficient procedure to find the probed edge subset E' and tree decomposition T ? While Bodlaender’s celebrated algorithm ([Bodlaender, 1996](#)) can efficiently find tree decompositions of constant width, it is unclear how to find the edge subset E' when E does not directly admit a decomposition of constant width. Approximation algorithms for this *interdiction* task exist ([Bansal et al., 2017](#)), but we do not know how to translate their approximation guarantees into statistical performance guarantees. Even if a set E' which admits a constant width decomposition is known to exist a-priori, we cannot guarantee that an approximation algorithm will not delete extra edges and consequently degrade the statistical recovery rate.

References

- Emmanuel Abbe, Afonso S Bandeira, Annina Bracher, and Amit Singer. Decoding binary node labels from censored edge measurements: Phase transition and efficient recovery. *Network Science and Engineering, IEEE Transactions on*, 1(1):10–22, 2014.
- Nikhil Bansal, Daniel Reichman, and Seeun William Umboh. Lp-based robust algorithms for noisy minor-free and bounded treewidth graphs. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1964–1979. SIAM, 2017.
- Hans L Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on computing*, 25(6):1305–1317, 1996.
- Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. Concentration inequalities using the entropy method. *Annals of Probability*, pages 1583–1614, 2003.
- Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Advanced lectures on machine learning*, pages 169–207. Springer, 2004.
- Yuxin Chen and Andrea J Goldsmith. Information recovery from pairwise measurements. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 2012–2016. IEEE, 2014.
- Yuxin Chen, Govinda Kamath, Changho Suh, and David Tse. Community recovery in graphs with locality. In *International Conference on Machine Learning*. International Machine Learning Society, 2016.
- Robert G Cowell, Philip Dawid, Steffen L Lauritzen, and David J Spiegelhalter. *Probabilistic networks and expert systems: Exact computational methods for Bayesian networks*. Springer Science & Business Media, 2006.
- Reinhard Diestel and Malte Müller. Connected tree-width. 2016.
- Jian Ding, Eyal Lubetzky, and Yuval Peres. Anatomy of the giant component: The strictly supercritical regime. *European Journal of Combinatorics*, 35:155–168, 2014.
- Abraham D Flaxman. Expansion and lack thereof in randomly perturbed graphs. *Internet Mathematics*, 4(2-3):131–147, 2007.
- Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.
- Amir Globerson, Tim Roughgarden, David Sontag, and Cafer Yildirim. Tight error bounds for structured prediction. *arXiv preprint arXiv:1409.5834*, 2014.
- Amir Globerson, Tim Roughgarden, David Sontag, and Cafer Yildirim. How hard is inference for structured prediction? *Proceedings of The 32nd International Conference on Machine Learning*, pages 2181–2190, 2015.
- Bruce Hajek, Yihong Wu, and Jiaming Xu. Computational lower bounds for community detection on random graphs. *arXiv preprint arXiv:1406.6625*, 2014.
- Thorsten Joachims and John E Hopcroft. Error bounds for correlation clustering. In *Proceedings of the 22nd International Conference on Machine Learning (ICML-05)*, pages 385–392, 2005.

- Jon Kleinberg and Eva Tardos. *Algorithm design*. Pearson Education India, 2006.
- Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Sciences*, 110(52):20935–20940, 2013.
- Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Correlation clustering with noisy partial information. In *Proceedings of The 28th Conference on Learning Theory*, pages 1321–1342, 2015.
- Elchanan Mossel and Jiaming Xu. Local algorithms for block models with side information. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 71–80. ACM, 2016.
- Mark EJ Newman and Duncan J Watts. Renormalization group analysis of the small-world network model. *Physics Letters A*, 263(4):341–346, 1999.
- Boris Pittel and Nicholas C Wormald. Counting connected graphs inside-out. *Journal of Combinatorial Theory, Series B*, 93(2):127–172, 2005.
- Neil Robertson and Paul D. Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986.
- Alaa Saade, Florent Krzakala, Marc Lelarge, and Lenka Zdeborova. Spectral detection in the censored block model. In *Information Theory (ISIT), 2015 IEEE International Symposium on*, pages 1184–1188. IEEE, 2015.
- James L Sharpnack, Akshay Krishnamurthy, and Aarti Singh. Near-optimal anomaly detection in graphs using lovasz extended scan statistic. In *Advances in Neural Information Processing Systems*, pages 1959–1967, 2013.
- David Sontag, Talya Meltzer, Amir Globerson, Tommi S Jaakkola, and Yair Weiss. Tightening lp relaxations for map using message passing. *arXiv preprint arXiv:1206.3288*, 2012.

A Further discussion of related work

Censored Block Model A recent line of research has studied recovery under the so-called *censored block model* (CBM). In CBM, vertices are labeled by ± 1 and for every edge uv , the number $Y_u Y_v$ is observed independently with probability $1 - q$ (where Y_u, Y_v are the labels of the vertices). The goal is to find the true label $Y_u Y_v$ of each edge uv correctly with high probability (based on the noisy observations). For partial recovery in the censored block model we ask for a prediction whose correlation with the ground truth (up to sign) is constant strictly greater than $1/2$ as $n \rightarrow \infty$. For the Erdős-Rényi random graph model, $G(n, \alpha/n)$ both the threshold (how large α needs to be in terms of p) for partial [Saade et al. \(2015\)](#) and exact [Abbe et al. \(2014\)](#) recovery have been determined. Exact recovery is obtained through maximum likelihood estimation which is generally intractable. The authors provide a polynomial time algorithm based on semidefinite programming that matches this threshold up to constant factors.

We observe that in our setting, due to the presence of side information, there is a simple and efficient algorithm that achieves exact recovery with high probability when the minimal degree is $\Omega(\log n)$: [Theorem 6](#). Such exact recovery algorithms are known for CBM model only under additional spectral expansion conditions [Abbe et al. \(2014\)](#).

Recovery from Pairwise Measurements [Chen and Goldsmith \(2014\)](#) also provides conditions on exact recovery in a censored block model-like setting which, like our own, considers structured classes of graphs. Motivated by applications in computational biology and social networks analysis, [Chen et al. \(2016\)](#) have recently considered *exact recovery* for edges in this setting for sparse graphs such as grid and rings. They consider the case where there are *multiple* i.i.d observations of edge labels. In contrast, we focus on the case where there is a single (noisy) observation for each edge.

Correlation Clustering Correlation clustering focuses on a combinatorial optimization problem closely related to the maximum likelihood estimation problem for our setting when we are only given edge labels. The main difference from our work is that the number of clusters is not predetermined. Most work on this setting has focused on obtaining approximation algorithms and has not considered any particular generative model for the weights (as in our case). An exception is [Joachims and Houghton \(2005\)](#), which gives partial recovery results in a model similar to the one we consider, in which a ground truth partition is fixed and the observed edge labels correspond to some noisy notion of similarity. However, these authors focus on the case where G is the complete graph.

[Makarychev et al. \(2015\)](#) consider correlation clustering where the model is a semi-random variant of the one we consider for the edge inference problem: Fix a graph $G = (V, E)$ and a vertex label Y . For each $uv \in E$, we observe X_{uv} where $X_{uv} = Y_u Y_v$ with probability $1 - p$ and has its value selected by an adversary otherwise. They do not consider side information, nor are they interested in concrete structured classes of graphs like grids.

Anomaly Detection [Sharpnack et al. \(2013\)](#) considered anomaly detection on graphs using only vertex measurements. Here it was also observed that constant correlation in vertex measurements leads to near-optimal recovery rates on structured classes of graphs.

B Omitted Proofs

B.1 Proofs from [Section 2](#)

Proof of [Theorem 1](#). By the Bernstein inequality we have that with probability at least $1 - \delta/2$,

$$\sum_{(u,v) \in E} \mathbb{1}\{Y_u \neq X_{u,v} Y_v\} \leq 2pn + 2 \log(2/\delta).$$

Thus, if we take $\mathcal{F} = \left\{ \hat{Y} : \sum_{(u,v) \in E} \mathbb{1}\{\hat{Y}_u \neq X_{u,v} \hat{Y}_v\} \leq 2pn + 2 \log(2/\delta) \right\}$ we will have $Y \in \mathcal{F}$ with probability at least $1 - \delta/2$.

Fix $\hat{Y} \in \{\pm 1\}^V$. We can verify by substitution that for each $v \in V$, $\mathbb{1}\{\hat{Y}_v \neq Y_v\} = \frac{1}{1-2q} \left[\mathbb{P}_Z(\hat{Y}_v \neq Z_v) - \mathbb{P}_Z(Y_v \neq Z_v) \right]$, so when $Y \in \mathcal{F}$ we have the following relation for Hamming error:

$$\sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v\} = \frac{1}{1-2q} \left[\sum_{v \in V} \mathbb{P}(\hat{Y}_v \neq Z_v) - \min_{Y' \in \mathcal{F}} \sum_{v \in V} \mathbb{P}(Y'_v \neq Z_v) \right].$$

Corollary 2 now implies that if we take $\hat{Y} = \arg \min_{Y' \in \mathcal{F}} \sum_{v \in V} \mathbb{1}\{Y'_v \neq Z_v\}$, which is precisely the solution to (2), we will have that with probability at least $1 - \delta/2$,

$$\sum_{v \in V} \mathbb{P}(\hat{Y}_v \neq Z_v) - \min_{Y' \in \mathcal{F}} \sum_{v \in V} \mathbb{P}(Y'_v \neq Z_v) \leq \left(\frac{4}{3} + \frac{1}{\epsilon} \right) \log \left(\frac{2|\mathcal{F}|}{\delta} \right).$$

Using that $|\mathcal{F}| \leq \sum_{k=0}^{2pn+2\log(2/\delta)} \binom{n}{k} \leq (e/p)^{2pn+2\log(2/\delta)}$ and $\epsilon \leq 1/2$ we further have that the RHS is bounded as $\frac{2}{\epsilon} \log(2e/p\delta)(2pn+2\log(2/\delta)+1)$. Putting everything together (and recalling $1-2q=2\epsilon$), we see that with probability at least $1-\delta$

$$\sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v\} \leq \frac{1}{\epsilon^2} (2pn+2\log(2/\delta)+1) \log(2e/p\delta).$$

□

B.2 Proofs from Section 3

Proof of Theorem 3. The minimax value of the estimation problem is given by

$$\min_{\hat{Y}} \max_Y \mathbb{E}_{X,Z|Y} \sum_{v \in V} \mathbb{1}\{\hat{Y}_v(X, Z) \neq Y_v\}.$$

We can move to a lower bound by considering a game where each vertex predictor \hat{Y}_v is given access to the true labels Y of all other vertices in G :

$$\geq \min_{\{\hat{Y}_v\}_{v \in V}} \max_Y \mathbb{E}_{X,Z|Y} \sum_{v \in V} \mathbb{1}\{\hat{Y}_v(X, Z, Y_{V \setminus \{v\}}) \neq Y_v\}.$$

Under the new model, the minimax optimal predictor for a given node v is given by the MAP predictor:

$$\hat{Y}_v = \arg \min_{\hat{Y} \in \{\pm 1\}} \log\left(\frac{1-q}{q}\right) \mathbb{1}\{\hat{Y} \neq Z_v\} + \log\left(\frac{1-p}{p}\right) \sum_{u \in N_v} \mathbb{1}\{\hat{Y} \neq Y_u X_{uv}\}.$$

When $p < q$, the minimax optimal estimator for v takes the majority of the predictions suggested by its edges (that is, $Y_u \cdot X_{uv}$ for each neighbor u) and uses the vertex observation Z_v to break ties.

When $\deg(v)$ is odd, the majority will be wrong if at least $\lceil \deg(v)/2 \rceil$ of the edges in the neighbor of v are flipped, and will be correct otherwise. When $\deg(v)$ is even there are two cases: 1) Strictly more than $\lceil \deg(v)/2 \rceil$ of the edges in $N(v)$ have been flipped, in which case the majority will be wrong. 2) Exactly half the edges are wrong, in which the optimal estimator will take the label Z_v as its prediction, which will be wrong with probability q . We thus have

$$\begin{aligned} \mathbb{P}(\hat{Y}_v \neq Y_v) &= \sum_{k=\lceil \deg(v)/2 \rceil}^{\deg(v)} \binom{\deg(v)}{k} p^k (1-p)^{\deg(v)-k} \\ &\geq \binom{\deg(v)}{\lceil \deg(v)/2 \rceil} p^{\lceil \deg(v)/2 \rceil} (1-p)^{\deg(v)-\lceil \deg(v)/2 \rceil} \\ &\geq \left(\frac{\deg(v)}{\lceil \deg(v)/2 \rceil} \right)^{\lceil \deg(v)/2 \rceil} p^{\lceil \deg(v)/2 \rceil} (1/2)^{\lceil \deg(v)/2 \rceil} \\ &\geq \Omega(p^{\lceil \deg(v)/2 \rceil}). \end{aligned}$$

In the last line we have used that we treat $\deg(v)$ as constant to suppress a weak dependence on it that arises when $\deg(v)$ is odd. Putting everything together, we see that in expectation we have the bound

$$\mathbb{E} \left[\sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v\} \right] \geq \Omega \left(q \sum_{v \in V} p^{\lceil \deg(v)/2 \rceil} \right).$$

□

Proof of Theorem 4. Recall that the minimax value of the estimation problem is given by

$$\min_{\hat{Y}} \max_Y \mathbb{E}_{X, Z|Y} \sum_{v \in V} \mathbb{1}\{\hat{Y}_v(X, Z) \neq Y_v\}.$$

As in the proof of Theorem 3, we will move to a lower bound where predictors are given access to extra data. In this case, we consider a set of disjoint predictors $\{\hat{Y}^W\}$, one for each component $W \in \mathcal{W}$. We assume that \hat{Y}^W see the ground truth Y_v for each vertex $v \notin W$, and further sees the product $Y_{uv} \triangleq Y_u Y_v$ for each edge $e \in E(W)$. Assuming $G(W)$ is connected (this clearly can only make the problem easier), the learner now only needs to infer one bit of information per component. The minimax value of the new game can be written as:

$$\geq \min_{\{\hat{Y}^W\}_{W \in \mathcal{W}}} \max_Y \mathbb{E}_{X, Z|Y} \sum_{W \in \mathcal{W}} \sum_{v \in W} \mathbb{1}\{\hat{Y}_v^W(X, Z, Y_{V \setminus W}, \{Y_{uv} \mid uv \in E(W)\}) \neq Y_v\}.$$

Because the learner only needs to infer a single bit per component, we have reduced to the setting of Theorem 3, components in our setting as vertices in that setting (so $\deg(v)$ is replaced by $\delta_G(W)$). The only substantive difference is the following: In that lower bound, we required that $p < q$. For the new setting, we have that “ q ” is actually (pessimistically) $q^{|W|}$, and so we require that $p < q^{\max_{W \in \mathcal{W}} |W|}$ for the bound to apply across all components. Using the final bound from Theorem 3, we have

$$\mathbb{E} \left[\sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v\} \right] \geq \Omega \left(q^{\max_{W \in \mathcal{W}} |W|} \sum_{W \in \mathcal{W}} p^{\lceil \delta_G(W)/2 \rceil} \right).$$

□

B.3 Proofs from Section 4

Proof of Example 1. We will show that $\Omega(pn)$ Hamming error is optimal for all trees by establishing that all trees have constant fraction of vertices whose degree is at most two, then appealing to Theorem 3.

Let T be the tree under consideration. T is bipartite. Let (A, B) be the bipartition of T into two disjoint independent sets. Suppose without loss of generality that $|A| \geq n/2$. If a is the number of vertices in A of degree at least 3 and $a' = |A| - a$, we have that $3a \leq n - 1$, hence $a \leq (n - 1)/3$. Therefore $a' \geq n/2 - a \geq (n - 1)/6$. Letting A' be the set of vertices in A with at most 2 neighbors, we see that A' is an independent set of size at least $(n - 1)/6$, and so we appeal to Theorem 3 for the result. □

Proof of Example 3. Fix $d \geq 3$. We will construct a graph G of size $(d + 1)n$. By building up from components as follows:

- For each $k \in [n]$ let G_k be the complete graph on $d + 1$ vertices. Remove an edge from an arbitrary pair of vertices (u_k, v_k) .
- Form G by taking the collection of all G_k , then adding an edge connecting v_k to u_{k+1} for each k , with the convention $u_{n+1} = u_1$.

This construction for $d = 3$ is illustrated in Figure 1.

Observe that G is d -regular. We obtain the desired result by applying Theorem 4 with the collection $\{G_k\}$ as the set system and observing that the each component G_k has only two edges leaving.

□

Proof of Example 4. We first examine the case where $c = 3$. Here we take the tree decomposition illustrated in Figure 2, where we cover the graph with overlapping 3×2 components, and take $W^* = \bigcup_{v \in W} N_v$. This yields $\text{mincut}^*(W) = 3$ for all components except those at the graph's endpoints. We now connect the components as a path graph and appeal to Theorem 2, which implies a rate of $\tilde{O}(p^2 n)$.

When $c = \omega(1)$ we can build a decomposition as follows (informally): Produce E' as in Figure 3 by performing the zig-zag cut with every third row of edges, leaving only 3 edges on the left or right side (alternating). We can now produce T (a line graph) by tiling G' with overlapping 3×3 components. Again, take $W^* = \bigcup_{v \in W} N_v$.

We can verify that if we perform extended inference we have $\text{mincut}^*(W) = 3$ for the $O(n)$ components in the interior of the graph and $\text{mincut}^*(W) = 2$ for the $O(\sqrt{n})$ components at the boundary.

The tree decomposition is illustrated in Figure 4. We have $\text{wid}^*(T) = O(1)$ and $\deg_E(T) = O(1)$. Applying Theorem 2 thus gives an upper bound of $\tilde{O}(p^2 n + p\sqrt{n})$ with probability at least $1 - \delta$.

Since T is a line graph, we pay $O(n \lceil p^2 n \rceil)$ in computation as per Appendix D.

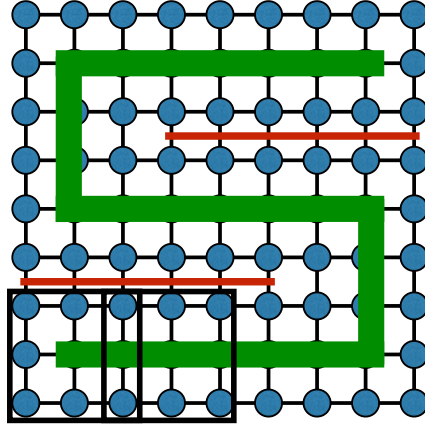


Figure 4: Tree decomposition for 2D grid.

□

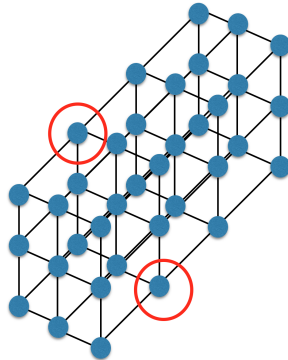


Figure 5: Lower bound argument for $n/c^2 \times c \times c$ hypergrid.

Proof of Example 5. We will prove this result for the three-dimensional case. We first show the lower bound.

Suppose $c \geq 3$ is constant, so that we are in the “hypertube” regime. Note that vertices on the outermost “edges” of the hypertube, examples of which are circled in Figure 5, have degree at most 4. There are $\Omega(n)$ such vertices, so appealing to Theorem 3 yields a lower bound on Hamming error of $\Omega(p^2n)$. In fact for the $n/c^2 \times c \times c$ hyper-tube one can achieve the $O(p^2n)$ rate using our method. Simply take each components of size $2 \times c \times c$ connected in a path as in the example for the 2D grid. Since the minimum cut for each component is already at least 3, we don’t need to consider extended components and simply use brute-force on the components themselves.

We now sketch the upper bound for the $n^{1/3} \times n^{1/3} \times n^{1/3}$ hypergrid. We use a technique similar to that used for the 2D grid in Example 4: We take T to be a line graph obtained by covering the hypergrid in overlapping $3 \times 3 \times 3$ components in a zig-zagging pattern. Note that each $3 \times 3 \times 3$ component will contain nodes similar to those highlighted in Figure 5 with degree at most 4. This means $\text{mincut}^*(W) = 4$, so to obtain the $O(p^3n)$ Hamming error we must consider extended components. Take $W^* = \bigcup_{v \in W} N_v$. Then $\text{mincut}^*(W) = 6$ for all components except those at the boundary of the hypergrid, which have $\text{mincut}^*(W) \in \{4, 5\}$. There are only $o(n)$ such components, so we achieve the $O(p^3n)$ upper bound by appealing to Theorem 2.

For higher-dimensional hypergrids, the strategy of taking components to be constant-sized hypergrids and T to be a zig-zagging line graph readily extends. The lower bound stated follows from a simple counting argument.

In general, we can associated vertices of a $c_1 \times c_2 \times \dots \times c_d$ hypergrid with the elements of $\mathbb{Z}_{c_1} \times \mathbb{Z}_{c_2} \times \dots \times \mathbb{Z}_{c_d}$. For a vertex $v = (v_1, \dots, v_d)$, the degree is given by $\deg(v) = |\{k \in [d] \mid v_k \in \{0, c_k\}\}|$.

Consider the case where $c_1, \dots, c_{d-1} = c$, $c_d = n/c^{d-1}$. In this case, the degree argument above implies

$$|\{v \mid \deg(v) = d + 1\}| \geq \sum_{v_k \in \{0, c\}: k \neq d} (n - 2) = \Omega(n).$$

Thus, a constant fraction of vertices have degree $d + 1$, and so Theorem 3 implies a lower bound of $\Omega(p^{\lceil \frac{d+1}{2} \rceil} n)$. □

Proof of Example 8.

Upper bound: Tree decomposition We first formally define the tree decomposition $T = (\mathcal{W}, F)$ that we will use with Algorithm 1. Assume for simplicity what $n = n' \cdot (2k + 1)$. We will define a vertex set $\{v_1, \dots, v_{n'}\}$ as follows: $v_1 = 1$, $v_{i+1} = v_i + k + 1$. We will now define a component for each of these vertices:

$$W(v_i) = N_G(v_i).$$

Let \mathcal{W} will be the union of these components. Since we assumed n to be divisible by $(2k + 1)$, the components a partition of V . We now define the EXTEND function for this decomposition:

$$\text{EXTEND}(W) = \bigcup_{v \in W} N_G(v).$$

That is, the extended component $W^*(v_i)$ is the set of all vertices removed from v_i by paths of length 2.

Finally, we construct the edge set F by adding edges of the form $(W(v_i), W(v_{i+1}))$ for $i \in \{1, \dots, n' - 1\}$. This means that the decomposition is a line graph. The decomposition is clearly admissible in the sense of Definition 3.

We can observe that $\text{mincut}^*(W) = 2k$ just as the minimum cut of $R_{n,k}$ is itself $2k$. [Theorem 2](#) thus implies a recovery rate of $\tilde{O}(p^k n)$. Since T is a line graph, the algorithm runs in time $O(\lceil p^k n \rceil n)$.

Lower bound That $O(p^k n)$ is optimal can be seen by appealing to [Theorem 3](#) with the fact that $R_{n,k}$ is $2k$ -regular. □

Proof of [Example 9](#). The average number of vertices added is αn . By the Chernoff bound, with high probability the number of vertices added is bounded as $\alpha n + c\sqrt{\alpha n \log n}$ for some constant c . This means that for any $\epsilon > 0$, there is some minimum n for which an $(1 - \alpha + \epsilon)$ fraction of vertices have no edges added. This means that there are at least $(1 - \alpha + \epsilon)n$ edges with degree $2k$, so [Theorem 3](#) yields the result. □

B.4 Analysis of TreeDecompositionDecoder

Properties of Tree Decompositions We begin by recalling a few properties of tree decompositions that are critical for proving the performance bounds for [Algorithm 1](#).

Proposition 2. For any tree decomposition $T = (\mathcal{W}, F)$, the following properties hold:

1. For each $v \in V$ there exists W with $v \in W$.
Guarantees that we produce a prediction for each vertex.
2. If $(W_1, W_2) \in F$, there is some $v \in V$ with $v \in W_1, W_2$.
Guarantees that the class \mathcal{F} (see [\(20\)](#)) is well-defined.
3. T is connected
So that $|\mathcal{F}| \lesssim 2^K$.
4. $|\mathcal{W}| \leq n$.
So that a mistake bound for components of the tree decomposition translates to a mistake bound for vertices of G .

Proof of [Proposition 2](#). 1. [Definition 1](#).

2. Suppose there is some edge $(W_1, W_2) \in F$ with no common vertices. Consider the subtrees T_1 and T_2 created by removing $(W_1, W_2) \in F$. By the coherence property ([Definition 1](#)), the subgraphs of G' associated with these decompositions (call them G'_{T_1} and G'_{T_2}) must have no common nodes. Yet, G' is connected, so there must be $(u, v) \in E'$ with $u \in G'_{T_1}$, $v \in G'_{T_2}$. Our hypothesis now implies that there is no $W \in \mathcal{W}$ containing u and v , so T violates the edge inclusion property of the tree decomposition.

3. [Definition 1](#)

4. This follows directly from the non-redundancy assumption of [Definition 1](#). See, e.g., ([Kleinberg and Tardos, 2006](#), 10.16). □

Estimation in Tree Decomposition Components We now formally define and analyze the component-wise estimators computed in [Line 6](#) of [Algorithm 1](#).

Definition 6 (Extended Component Estimator). *Consider the (edge) maximum likelihood estimator over W^* :*

$$\tilde{Y}^{W^*} \triangleq \arg \min_{\tilde{Y} \in \{\pm 1\}^{W^*}} \sum_{uv \in E'(W^*)} \mathbb{1}\{\tilde{Y}_u \tilde{Y}_v \neq X_{uv}\}. \quad (12)$$

We define the **extended component estimator** $\hat{Y}^{W^*} \in \{\pm 1\}^W$ as restriction of \tilde{Y}^{W^*} to W .

For \hat{Y}^{W^*} estimation performance is governed by $\text{mincut}^*(W)$ rather than $\text{mincut}(W)$, as the next lemma shows:

Lemma 2 (Error Probability for Extended Component Estimator).

$$\mathbb{P}\left(\min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^{W^*} \neq Y^W\} > 0\right) \leq 2^{|W^*|} p^{\lceil \text{mincut}^*(W)/2 \rceil}.$$

Proof of Lemma 2. Suppose $\hat{Y}^{W^*} \neq Y^W$ and consider $D = \{v \in W^* : \tilde{Y}_v^{W^*} \neq Y_v\}$. Then there is some maximal connected component S of D containing at least one vertex of W . It must then be the case that at least half the edge samples in $\delta(S)$ are flipped with respect to the ground truth. Subsequently we have the bound

$$\begin{aligned} \mathbb{P}\left(\min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^{W^*} \neq Y^W\} > 0\right) &\leq \sum_{S \subseteq W^* : S \cap W \neq \emptyset, \bar{S} \cap W \neq \emptyset} p^{\lceil |\delta(S)|/2 \rceil} \\ &\leq \sum_{S \subseteq W^*} p^{\lceil \text{mincut}^*(W)/2 \rceil} \\ &\leq 2^{|W^*|} p^{\lceil \text{mincut}^*(W)/2 \rceil}. \end{aligned}$$

□

[Lemma 2](#) shows that considering mincut^* offers improved failure probability over mincut because it allows us to take advantage of all of the information in W^* , yet only pay (in terms of errors) for cuts that involve nodes in the core component W . In [Figure 2](#), all components of the tree decomposition except the endpoints have $\text{mincut}^*(W) = 3$, and so their extended component estimators achieve $O(p^2)$ failure probability.

Concentration We begin by stating a concentration result for functions of independent random variables, which we will use to establish a bound on the total number of components that fail in the first stage of our algorithm. Let X_1, \dots, X_n be independent random variables each taking values in a probability space \mathcal{X} , and let $F : \mathcal{X}^n \rightarrow \mathbb{R}$. We will be interested in the concentration of the random variable $S = F(X_1, \dots, X_n)$. Letting X'_1, \dots, X'_n be independent copies of X_1, \dots, X_n , we define $S^{(i)} = F(X_1, \dots, X_{i-1}, X'_i, X_{i+1}, \dots, X_n)$. Finally, we define a new random variable

$$V_+ = \sum_{i=1}^n \mathbb{E}\left[(S - S^{(i)})_+^2 \mid X_1, \dots, X_n\right].$$

Theorem 5 (Entropy Method with Efron-Stein Variance ([Boucheron et al., 2003](#))). *If there exists a constant $a > 0$ such that $V_+ \leq aS$ then*

$$\mathbb{P}\{S \geq \mathbb{E}[S] + t\} \leq \exp\left(\frac{-t^2}{4a \mathbb{E}[S] + 2at}\right).$$

Subsequently, with probability at least $1 - \delta$,

$$S \leq \mathbb{E}[S] + \max\left\{4a \log(1/\delta), 2\sqrt{2a \mathbb{E}[S] \log(1/\delta)}\right\} \leq 2 \mathbb{E}[S] + 6a \log(1/\delta).$$

With [Theorem 5](#) in mind, we may proceed to a bound on the number of components with mistakes when the basic component estimator [\(9\)](#) is used.

Lemma 3 (Formal Version of [Lemma 1](#)). For all $\delta > 0$, with probability at least $1 - \delta$ over the draw of X ,

$$\min_{s \in \{\pm 1\}^{\mathcal{W}}} \sum_{W \in \mathcal{W}} \mathbb{1}\{s_W \hat{Y}^W \neq Y^W\} \leq 2 \sum_{W \in \mathcal{W}} 2^{|W|} p^{\lceil \text{mincut}(W)/2 \rceil} + 6 \max_{e \in E} |\mathcal{W}(e)| \max_{W \in \mathcal{W}} |E'(W)| \log(1/\delta). \quad (13)$$

$$(14)$$

Proof of Lemma 3. Define a random variable

$$S(X) = \sum_{W \in \mathcal{W}} \min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^W(X) \neq Y^W\},$$

where \hat{Y}^W are the component-wise estimators produced by [Algorithm 1](#) and X are the edge observations. To prove the lemma we will apply [Theorem 5](#) by showing that there is a constant a such that the necessary variance bound $V_+ \leq aS$ holds.

To this end, consider

$$S(X) - S(X^{(e)}) = \sum_{W \in \mathcal{W}} \left(\min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^W(X) \neq Y^W\} - \min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^W(X^{(e)}) \neq Y^W\} \right),$$

where $X^{(e)}$ is defined as in [Theorem 5](#). To be more precise, we draw $(X'_e)_{e \in E}$ from the same distribution as X , then let $X^{(e)}$ be the result of replacing X_e with X'_e .

We have

$$S(X) - S(X^{(e)}) = \sum_{W \in \mathcal{W}(e)} \left(\min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^W(X) \neq Y^W\} - \min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^W(X^{(e)}) \neq Y^W\} \right),$$

since changing X_e can only change \hat{Y}^W if $e \in W$. Now, since $S(X^{(e)})$ is nonnegative we have

$$\begin{aligned} (S(X) - S(X^{(e)}))_+^2 &= \left(\sum_{W \in \mathcal{W}(e)} \left(\min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^W(X) \neq Y^W\} - \min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^W(X^{(e)}) \neq Y^W\} \right) \right)_+^2 \\ &\leq \left(\sum_{W \in \mathcal{W}(e)} \min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^W(X) \neq Y^W\} \right)^2 \\ &\leq |\mathcal{W}(e)| \sum_{W \in \mathcal{W}(e)} \min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^W(X) \neq Y^W\}. \end{aligned}$$

We now sum over all edges to arrive at an upper bound on V_+ :

$$\begin{aligned}
V_+ &= \sum_{e \in E} \mathbb{E} \left[(S(X) - S(X^{(e)}))_+^2 \mid X \right] \\
&\leq \max_{e \in E} |\mathcal{W}(e)| \sum_{e \in E} \sum_{W \in \mathcal{W}(e)} \min_{s \in \{\pm 1\}} \mathbb{1} \left\{ s \hat{Y}^W(X) \neq Y^W \right\} \\
&= \max_{e \in E} |\mathcal{W}(e)| \sum_{W \in \mathcal{W}} \sum_{e \in E(W)} \min_{s \in \{\pm 1\}} \mathbb{1} \left\{ s \hat{Y}^W(X) \neq Y^W \right\} \\
&\leq \max_{e \in E} |\mathcal{W}(e)| \max_{W \in \mathcal{W}} |E(W)| \sum_{W \in \mathcal{W}} \min_{s \in \{\pm 1\}} \mathbb{1} \left\{ s \hat{Y}^W(X) \neq Y^W \right\} \\
&\leq \max_{e \in E} |\mathcal{W}(e)| \max_{W \in \mathcal{W}} |E(W)| \sum_{W \in \mathcal{W}} \min_{s \in \{\pm 1\}} \mathbb{1} \left\{ s \hat{Y}^W(X) \neq Y^W \right\} \\
&= \max_{e \in E} |\mathcal{W}(e)| \max_{W \in \mathcal{W}} |E(W)| S(X).
\end{aligned}$$

We now appeal to [Theorem 5](#) with $a = \max_{e \in E} |\mathcal{W}(e)| \max_{W \in \mathcal{W}} |E(W)|$, which yields that with probability at least $1 - \delta$

$$S \leq 2 \mathbb{E}[S] + 6 \max_{e \in E} |\mathcal{W}(e)| \max_{W \in \mathcal{W}} |E(W)| \log(1/\delta).$$

Finally, the bound on $\mathbb{E}[S]$ follows from [Proposition 1](#):

$$\mathbb{E}[S] = \sum_{W \in \mathcal{W}} \mathbb{P} \left(\min_{s \in \{\pm 1\}} \mathbb{1} \left\{ s \hat{Y}^W(X) \neq Y^W \right\} \right) \leq \sum_{W \in \mathcal{W}} 2^{|W|} p^{\lceil \text{mincut}(W)/2 \rceil}.$$

□

An analogous concentration result to [Lemma 3](#) holds to bounds the number of components that fail over the whole graph when the extended component estimator is used:

Lemma 4. For all $\delta > 0$, with probability at least $1 - \delta$ over the draw of X ,

$$\min_{s \in \{\pm 1\}} \sum_{W \in \mathcal{W}} \mathbb{1} \{ s_W \hat{Y}^{W^*} \neq Y^{W^*} \} \leq 2 \sum_{W \in \mathcal{W}} 2^{|W^*|} p^{\lceil \text{mincut}^*(W)/2 \rceil} + 6 \max_{e \in E} |\mathcal{W}^*(e)| \max_{W \in \mathcal{W}} |E'(W^*)| \log(1/\delta). \quad (15)$$

Proof of Lemma 4. This proof proceeds exactly as in the proof of [Lemma 3](#) using

$$S(X) = \sum_{W \in \mathcal{W}} \min_{s \in \{\pm 1\}} \mathbb{1} \left\{ s \hat{Y}^{W^*}(X) \neq Y^{W^*} \right\}.$$

The only difference is that edges are more influential than in that lemma because each extended component estimator \hat{Y}^{W^*} may depend on more edges than the simpler component estimator \hat{Y}^W . To this end, define $\mathcal{W}^*(e) = \{W \mid e \in E'(W^*)\}$. One can verify that if we replace every instance of $\mathcal{W}(e)$ in the proof of [Lemma 3](#) with $\mathcal{W}^*(e)$ it holds that $V_+ \leq aS$ with $a = \max_{e \in E} |\mathcal{W}^*(e)| \max_{W \in \mathcal{W}} |E(W^*)|$. [Theorem 5](#) then implies that with probability at least $1 - \delta$,

$$\begin{aligned}
S &\leq 2 \mathbb{E}[S] + 6 \max_{e \in E} |\mathcal{W}^*(e)| \max_{W \in \mathcal{W}} |E(W^*)| \log(1/\delta) \\
&= 2 \mathbb{E}[S] + 6 \deg_E^*(T) \max_{W \in \mathcal{W}} |E(W^*)| \log(1/\delta).
\end{aligned}$$

□

Proof of Theorem 2.

Full theorem statement We will prove the following error bound: If $T = (\mathcal{W}, F)$ is admissible, with probability at least $1 - \delta$ over the draw of X and Z , \hat{Y} satisfies:

$$\sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v\} \quad (16)$$

$$\leq O\left(\frac{1}{\epsilon^2} \left(2^{\text{wid}^*(T)} \sum_{W \in \mathcal{W}} p^{\lceil \text{mincut}^*(W)/2 \rceil} + \deg_E^*(T) \max_{W \in \mathcal{W}} |E(W^*)| \log(1/\delta)\right) \cdot (\text{wid}(T) + \deg(T) \log n)\right) \quad (17)$$

This statement specializes to (7) when all of the tree decomposition quantities are constant and $\delta = 1/n$.

Error bound for individual components Lemma 2 implies that for a fixed component $W \in \mathcal{W}$, the probability that the estimator produced by the brute-force enumeration routine fails to exactly recover the labels in W (up to sign) is bounded as

$$\mathbb{P}\left(\min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^{W^*} \neq Y^W\} > 0\right) \leq 2^{|W^*|} p^{\lceil \text{mincut}^*(W)/2 \rceil}.$$

Error bound across all components Consider the following random variable, which is the total number components

$$S(X) = \sum_{W \in \mathcal{W}} \min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^{W^*}(X) \neq Y^W\}.$$

The bound on component failure probability immediately implies in in-expectation bound on S :

$$\mathbb{E}[S] \leq \sum_{W \in \mathcal{W}} 2^{|W^*|} p^{\lceil \text{mincut}^*(W)/2 \rceil}.$$

Lemma 4 shows that S concentrates tightly around its expectation. More precisely, let $A = 6\deg_E^*(T) \max_{W \in \mathcal{W}} |E(W^*)|$ and

$$K_n = 2^{\text{wid}^*(T)+2} \sum_{W \in \mathcal{W}} p^{\lceil \text{mincut}^*(W)/2 \rceil} + A \log(2/\delta). \quad (18)$$

Then Lemma 4 implies that with probability at least $1 - \delta/2$,

$$\begin{aligned} \min_{s \in \{\pm 1\}^{\mathcal{W}}} \sum_{W \in \mathcal{W}} \mathbb{1}\{s_W \hat{Y}^{W^*} \neq Y^W\} &\leq 2 \sum_{W \in \mathcal{W}} 2^{|W^*|} p^{\lceil \text{mincut}^*(W)/2 \rceil} + A \log(2/\delta) \\ &\leq K_n \end{aligned} \quad (19)$$

Inference with side information: Hypothesis class Consider the following binary signing of the components in T :

$$s^* = \arg \min_{s \in \{\pm 1\}^{\mathcal{W}}} \sum_{W \in \mathcal{W}} \mathbb{1}\{s_W \hat{Y}^{W^*} \neq Y^W\}.$$

s^* is signing of the component-wise predictions (\hat{Y}^{W^*}) that best matches the ground truth. If we knew the value of s^* we could use it to produce a vertex prediction with at most K_n mistakes.

Computing the s^* is information-theoretically impossible because we do not have access to Y , but we will show that the signing we produce using the side information Z is close.

Let $L_n = \deg(T) \cdot K_n$. We will argue that (19) implies that s^* lies in the class

$$\mathcal{F}(X) \triangleq \left\{ s \in \{\pm 1\}^{\mathcal{W}} \mid \sum_{(W_1, W_2) \in F} \mathbb{1}\{s_{W_1} \neq s_{W_2} \cdot S(W_1, W_2)\} \leq L_n \right\}. \quad (20)$$

First, consider the For loop on Algorithm 1, Line 12. Proposition 2 implies that $S(W_1, W_2)$ as defined in this loop is well-defined, because there always exists some $v \in W_1 \cap W_2$.

Second, consider the value of

$$\sum_{(W_1, W_2) \in F} \mathbb{1}\{s_{W_1}^* \neq s_{W_2}^* \cdot S(W_1, W_2)\} = \sum_{(W_1, W_2) \in F} \mathbb{1}\{s_{W_1}^* \neq s_{W_2}^* \cdot \hat{Y}_v^{W_1^*} \cdot \hat{Y}_v^{W_2^*}\}.$$

We can bound this quantity in terms of the number of components W for which

$$\min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^{W^*} \neq Y^W\} = 1.$$

Observe that if $\min_{s \in \{\pm 1\}} \mathbb{1}\{s \hat{Y}^{W^*} \neq Y^W\} = 0$ then there is some $\bar{s}_W \in \{\pm 1\}$ such that $\hat{Y}^{W^*} = \bar{s}_W Y^W$. If we take $s_W^* = \bar{s}_W$ in all the components with no errors, and choose the sign arbitrarily for others, we will have $\mathbb{1}\{s_{W_1}^* \neq s_{W_2}^* \cdot \hat{Y}_v^{W_1^*} \cdot \hat{Y}_v^{W_2^*}\} = 0$ whenever both W_1 and W_2 have no errors. Pessimistically, there are at most $L_n = \deg(T) \cdot K_n$ edges (W_1, W_2) where at least one of W_1 or W_2 has an error, and therefore (19) implies that with probability at least $1 - \delta/2$, $s^* \in \mathcal{F}$.

We conclude this discussion by showing that $|\mathcal{F}(X)|$ is small. Since by Proposition 2 T is connected, labelings of the edges of T are in one to one correspondence with labelings of the components. Consequently,

$$|\mathcal{F}(X)| \leq \sum_{k=0}^{K_n} \binom{|\mathcal{W}|}{k} \leq \left(\frac{e|\mathcal{W}|}{K_n} \right)^{K_n} \leq \left(\frac{en}{K_n} \right)^{K_n}. \quad (21)$$

The last inequality uses that, from Proposition 2, $|\mathcal{W}| \leq n$.

Final error bound for inference with side information We now use the properties of $\mathcal{F}(X)$ to derive an error bound for the prediction \hat{Y} . Recall from Algorithm 1 that \hat{Y} is defined in terms of

$$\hat{s} = \min_{s \in \mathcal{F}(X)} \sum_{W \in \mathcal{W}} \sum_{v \in W} \mathbb{1}\{s_W \hat{Y}_v^{W^*} \neq Z_v\}. \quad (22)$$

We reduce the analysis of the error rate of \hat{s} to analysis of excess risk in a manner that parallels the proof of Theorem 1, but is slightly more involved because the best predictor in \mathcal{F} does not perfectly

match the ground truth. Fix $\hat{s} \in \{\pm 1\}^{\mathcal{W}}$. For each component $W \in \mathcal{W}$ we have

$$\begin{aligned}
\sum_{v \in W} \mathbb{1}\{\hat{s}_W \hat{Y}_v^{W*} \neq Y_v\} &\leq \sum_{v \in W} \mathbb{1}\{\hat{s}_W \hat{Y}_v^{W*} \neq s_W^* \hat{Y}_v^{W*}\} + \sum_{v \in W} \mathbb{1}\{s_W^* \hat{Y}_v^{W*} \neq Y_v\} \\
&\leq \sum_{v \in W} \mathbb{1}\{\hat{s}_W \hat{Y}_v^{W*} \neq s_W^* \hat{Y}_v^{W*}\} + |W| \mathbb{1}\{s_W^* \hat{Y}^{W*} \neq Y^W\} \\
&= \frac{1}{1-2q} \sum_{v \in W: s_W^* \hat{Y}_v^{W*} = Y_v} \left(\mathbb{P}_Z \left(\hat{s}_W \hat{Y}_v^{W*} \cdot Z_v < 0 \right) - \mathbb{P}_Z \left(s_W^* \hat{Y}_v^{W*} \cdot Z_v < 0 \right) \right) \\
&\quad - \frac{1}{1-2q} \sum_{v \in W: s_W^* \hat{Y}_v^{W*} \neq Y_v} \left(\mathbb{P}_Z \left(\hat{s}_W \hat{Y}_v^{W*} \cdot Z_v < 0 \right) - \mathbb{P}_Z \left(s_W^* \hat{Y}_v^{W*} \cdot Z_v < 0 \right) \right) \\
&\quad + |W| \mathbb{1}\{s_W^* \hat{Y}_W \neq Y_W\}.
\end{aligned}$$

Now note that given that Z_v is drawn as a noisy version of Y_v , $\left| \mathbb{P}_Z \left(\hat{s}_W \hat{Y}_v^{W*} \cdot Z_v < 0 \right) - \mathbb{P}_Z \left(s_W^* \hat{Y}_v^{W*} \cdot Z_v < 0 \right) \right| = 1 - 2q$ and so

$$\begin{aligned}
&-\frac{1}{1-2q} \sum_{v \in W: s_W^* \hat{Y}_v^{W*} \neq Y_v} \left(\mathbb{P}_Z \left(\hat{s}_W \hat{Y}_v^{W*} \cdot Z_v < 0 \right) - \mathbb{P}_Z \left(s_W^* \hat{Y}_v^{W*} \cdot Z_v < 0 \right) \right) \\
&\leq 2 \sum_{v \in W} \mathbb{1}\{s_W^* \hat{Y}_v^{W*} \neq Y_v\} + \frac{1}{1-2q} \sum_{v \in W: s_W^* \hat{Y}_v^{W*} \neq Y_v} \left(\mathbb{P}_Z \left(\hat{s}_W \hat{Y}_v^{W*} \cdot Z_v < 0 \right) - \mathbb{P}_Z \left(s_W^* \hat{Y}_v^{W*} \cdot Z_v < 0 \right) \right) \\
&\leq 2|W| \mathbb{1}\{s_W^* \hat{Y}^{W*} \neq Y^W\} + \frac{1}{1-2q} \sum_{v \in W: s_W^* \hat{Y}_v^{W*} \neq Y_v} \left(\mathbb{P}_Z \left(\hat{s}_W \hat{Y}_v^{W*} \cdot Z_v < 0 \right) - \mathbb{P}_Z \left(s_W^* \hat{Y}_v^{W*} \cdot Z_v < 0 \right) \right).
\end{aligned}$$

We thus conclude that

$$\begin{aligned}
&\sum_{v \in W} \mathbb{1}\{\hat{s}_W \hat{Y}_v^{W*} \neq Y_v\} \\
&\leq 3|W| \mathbb{1}\{s_W^* \hat{Y}^{W*} \neq Y^W\} + \frac{1}{1-2q} \sum_{v \in W} \left(\mathbb{P}_Z \left(\hat{s}_W \hat{Y}_v^{W*} \cdot Z_v < 0 \right) - \mathbb{P}_Z \left(s_W^* \hat{Y}_v^{W*} \cdot Z_v < 0 \right) \right).
\end{aligned}$$

Summing over all the components $W \in \mathcal{W}$ we arrive at the bound

$$\begin{aligned}
&\sum_{W \in \mathcal{W}} \sum_{v \in W} \mathbb{1}\{\hat{s}_W \hat{Y}_v^{W*} \neq Y_v\} \\
&\leq 3 \left(\max_{W \in \mathcal{W}} |W| \right) \sum_{w \in \mathcal{W}} \mathbb{1}\{s_w^* \hat{Y}^{w*} \neq Y^w\} + \frac{1}{1-2q} \sum_{W \in \mathcal{W}} \sum_{v \in W} \left(\mathbb{P}_Z \left(\hat{s}_W \hat{Y}_v^{W*} \cdot Z_v < 0 \right) - \mathbb{P}_Z \left(s_W^* \hat{Y}_v^{W*} \cdot Z_v < 0 \right) \right) \\
&\leq 3 \left(\max_{W \in \mathcal{W}} |W| \right) K_n + \frac{1}{1-2q} \sum_{W \in \mathcal{W}} \sum_{v \in W} \left(\mathbb{P}_Z \left(\hat{s}_W \hat{Y}_v^{W*} \cdot Z_v < 0 \right) - \mathbb{P}_Z \left(s_W^* \hat{Y}_v^{W*} \cdot Z_v < 0 \right) \right)
\end{aligned}$$

We can now appeal to our statistical learning tools to bound the RHS of this expression. [Lemma 5](#) implies that if we take $\hat{s} = \arg \min_{s \in \mathcal{F}} \sum_{W \in \mathcal{W}} \sum_{v \in W} \mathbb{1}\{\hat{s}_W \hat{Y}_v^{W*} \cdot Z_v < 0\}$, which is precisely the solution to [\(22\)](#), we obtain the excess risk bound,

$$\begin{aligned}
&\sum_{W \in \mathcal{W}} \sum_{v \in W} \left(\mathbb{P}_Z \left(\hat{s}_W \hat{Y}_v^{W*} \cdot Z_v < 0 \right) - \mathbb{P}_Z \left(s_W^* \hat{Y}_v^{W*} \cdot Z_v < 0 \right) \right) \\
&\leq \left(\frac{2}{3} + \frac{c}{2} \right) \log(2|\mathcal{F}|/\delta) + \frac{1}{c} \sum_{w \in \mathcal{W}} \sum_{v \in W} \mathbb{1}\{\hat{s}_W \hat{Y}_v^{W*} \neq Y_v\},
\end{aligned}$$

with probability at least $1 - \delta/2$ over Z for all $c > 0$. If we choose $c = 1/\epsilon$, rearrange, and apply the union bound, this implies that with probability at least $1 - \delta$ over the draw of X and Z we have

$$\sum_{W \in \mathcal{W}} \sum_{v \in W} \mathbb{1}\{\hat{s}_W \hat{Y}_v^{W^*} \neq Y_v\} \leq 6 \left(\max_{W \in \mathcal{W}} |W| \right) K_n + \frac{2}{\epsilon^2} \log(2|\mathcal{F}|/\delta).$$

Recall that $|\mathcal{F}| \leq (e|\mathcal{W}|/L_n)^{L_n}$, which implies a bound of

$$\begin{aligned} & \sum_{W \in \mathcal{W}} \sum_{v \in W} \mathbb{1}\{\hat{s}_W \hat{Y}_v^{W^*} \neq Y_v\} \\ & \leq O\left(\frac{1}{\epsilon^2} [\text{wid}(T) \cdot K_n + L_n \cdot \log(en/L_n) + \log(1/\delta)]\right) \\ & \leq O\left(\frac{1}{\epsilon^2} [K_n \cdot (\text{wid}(T) + \deg(T) \cdot \log(en/K_n)) + \log(1/\delta)]\right) \\ & \leq O\left(\frac{1}{\epsilon^2} \left(2^{\text{wid}^*(T)} \sum_{W \in \mathcal{W}} p^{\lceil \text{mincut}^*(W)/2 \rceil} + \deg_E^*(T) \max_{W \in \mathcal{W}} |E(W^*)| \log(1/\delta) \right) \cdot (\text{wid}(T) + \deg(T) \log n) \right) \end{aligned}$$

Our choice of \hat{Y} in [Algorithm 1](#) ensures that the Hamming error $\sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v\}$ inherits this bound. [Proposition 2](#) implies that every $v \in V$ is in some component, so this choice is indeed well-defined. \square

C Statistical Learning Theory

Here we consider a *fixed design* version of the statistical learning setting. Fix an input space \mathcal{X} and output space \mathcal{Z} . We are given a fixed set $X_1, \dots, X_n \in \mathcal{X}$ and samples $Z_1, \dots, Z_n \in \mathcal{Z}$ with Z_i drawn from $P(Z_i | X_i)$ for some distribution P . We fix a hypothesis class \mathcal{F} which is some subset of mappings from \mathcal{X} to \mathcal{Z} , and we would like to use Z to find $\hat{Y} \in \mathcal{F}$ that will predict future observations of Z on X . To evaluate prediction we define a *loss function* $\ell : \mathcal{Z} \times \mathcal{Z} \rightarrow \mathbb{R}_+$, and define $L_i(Y) = \mathbb{E}_{Z|X_i}[\ell(Y, Z)]$. Our goal is to use Z to select $\hat{Y} \in \mathcal{F}$ to minimize the *excess risk*:

$$\sum_{i \in [n]} L_i(\hat{Y}(X_i)) - \min_{Y \in \mathcal{F}} \sum_{i \in [n]} L_i(Y(X_i)). \quad (23)$$

Typically this is accomplished using the *empirical risk minimizer* (ERM):

$$\hat{Y} = \arg \min_{Y \in \mathcal{F}} \sum_{i \in [n]} \ell(Y(X_i), Z_i)^4.$$

In this paper we consider a specific instantiation of the above framework in which

- $\mathcal{X} = V$, the vertex set for some graph (possibly a tree decomposition), and X_1, \dots, X_n are an arbitrary ordering of V (so $n = |V|$). In light of this we index all variables using V going forward.
- $\mathcal{Z} = \{\pm 1\}$. We fix $Y \in \{\pm 1\}^V$ and let $Z_v = Y_v$ with probability $1 - q$ and $Z_v = -Y_v$ otherwise (as in [Model 1](#)).

⁴There are many standard bounds quantifying the performance of ERM in settings beyond the one we consider. See [Bousquet et al. \(2004\)](#) for a survey.

- $\ell(Y, Z) = \mathbb{1}\{Y \neq Z\}$, so $L_i(Y) = \mathbb{P}_Z(Y \neq Z_v)$.
- $\mathcal{F} \subseteq \{\pm 1\}^V$ can be arbitrary.

For this setting the excess risk for a predictor $\hat{Y} \in \{\pm 1\}^V$ can be written as

$$\sum_{v \in V} \mathbb{P}(\hat{Y}_v \neq Z_v) - \min_{Y' \in \mathcal{F}} \sum_{v \in V} \mathbb{P}(Y'_v \neq Z_v), \quad (24)$$

and the empirical risk minimizer is given by $\hat{Y} = \arg \min_{Y' \in \mathcal{F}} \sum_{v \in V} \mathbb{1}\{Y'_v \neq Z_v\}$.

We assume this setting exclusively for the remainder of the section.

Lemma 5 (Excess Risk Bound for ERM). Let \hat{Y} be the ERM and let $Y^* = \arg \min_{Y' \in \mathcal{F}} \sum_{v \in V} \mathbb{P}(Y'_v \neq Z_v)$. Then with probability at least $1 - \delta$ over the draw of Z ,

$$\sum_{v \in V} \mathbb{P}(\hat{Y}_v \neq Z_v) - \min_{Y' \in \mathcal{F}} \sum_{v \in V} \mathbb{P}(Y'_v \neq Z_v) \leq \left(\frac{2}{3} + \frac{c}{2}\right) \log\left(\frac{|\mathcal{F}|}{\delta}\right) + \frac{1}{c} \sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v^*\} \quad (25)$$

for all $c > 0$.

Corollary 2 (ERM Excess Risk: Well-Specified Case). When $Y \in \mathcal{F}$ we have that with probability at least $1 - \delta$,

$$\sum_{v \in V} \mathbb{P}(\hat{Y}_v \neq Z_v) - \min_{Y \in \mathcal{F}} \sum_{v \in V} \mathbb{P}(Y_v \neq Z_v) \leq \left(\frac{4}{3} + \frac{1}{\epsilon}\right) \log\left(\frac{|\mathcal{F}|}{\delta}\right), \quad (26)$$

recalling $q = 1/2 - \epsilon$.

Proof of Corollary 2. When $Y \in \mathcal{F}$, $Y^* = Y$, and we have

$$\sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v\} = \frac{1}{1 - 2q} \sum_{v \in V} \left(\mathbb{P}(\hat{Y}_v \neq Z_v) - \mathbb{P}(Y_v \neq Z_v) \right).$$

Applying this inequality to the right hand side of (25) and rearranging yields

$$\left(1 - \frac{1}{c(1 - 2q)}\right) \sum_{v \in V} \left(\mathbb{P}(\hat{Y}_v \neq Z_v) - \mathbb{P}(Y_v \neq Z_v) \right) \leq \left(\frac{2}{3} + \frac{c}{2}\right) \log(|\mathcal{F}|/\delta).$$

To complete the proof we take $c = \frac{2}{1 - 2q}$, which gives

$$\frac{1}{2} \sum_{v \in V} \left(\mathbb{P}(\hat{Y}_v \neq Z_v) - \mathbb{P}(Y_v \neq Z_v) \right) \leq \left(\frac{2}{3} + \frac{1}{1 - 2q}\right) \log(|\mathcal{F}|/\delta).$$

□

Proof of Lemma 5. We will use Lemma 6 with \mathcal{F} as the index set so that every $i \in [N]$ corresponds to one $Y' \in \mathcal{F}$. We define our collection of random variables as

$$T_v^{Y'} = \mathbb{1}\{Y'_v \neq Z_v\} - \mathbb{1}\{Y_v^* \neq Z_v\}$$

where Y is the ground truth and Y' is any element of \mathcal{F} . Now using [Lemma 6](#) and recalling $\sigma_{Y'}^2 = \sum_{v \in V} \text{Var}(T_v^{Y'})$, we have that with probability at least $1 - \delta$, simultaneously for all Y' ,

$$\begin{aligned} \sum_{v \in V} (\mathbb{E}[T_v^{Y'}] - T_v^{Y'}) &\leq \frac{2}{3} \log(|\mathcal{F}|/\delta) + \sqrt{2\sigma_{Y'}^2 \log(|\mathcal{F}|/\delta)} \\ &\leq \inf_{c>0} \left[\left(\frac{2}{3} + \frac{c}{2} \right) \log(|\mathcal{F}|/\delta) + \sigma_{Y'}^2/c \right] \\ &\leq \inf_{c>0} \left[\left(\frac{2}{3} + \frac{c}{2} \right) \log(|\mathcal{F}|/\delta) + \frac{1}{c} \sum_{v \in V} \mathbb{E}[(T_v^{Y'})^2] \right]. \end{aligned}$$

In particular this implies that for $\hat{Y} = \arg \min_{Y' \in \mathcal{F}} \sum_{v \in V} \mathbb{1}\{Y'_v \neq Z_v\}$ we have that for all $c > 0$,

$$\begin{aligned} \sum_{v \in V} \left(\mathbb{P}(\hat{Y}_v \neq Z_v) - \mathbb{P}(Y_v^* \neq Z_v) \right) &\leq \sum_{v \in V} \left(\mathbb{1}\{\hat{Y}_v \neq Z_v\} - \mathbb{1}\{Y_v^* \neq Z_v\} \right) + \left(\frac{2}{3} + \frac{c}{2} \right) \log(|\mathcal{F}|/\delta) \\ &\quad + \frac{1}{c} \sum_{v \in V} \mathbb{E} \left[\left(\mathbb{1}\{\hat{Y}_v \neq Z_v\} - \mathbb{1}\{Y_v^* \neq Z_v\} \right)^2 \right]. \end{aligned}$$

Now since $Y^* \in \mathcal{F}$ and \hat{Y} is the ERM, we get that $\sum_{v \in V} \left(\mathbb{1}\{\hat{Y}_v \neq Z_v\} - \mathbb{1}\{Y_v^* \neq Z_v\} \right) \leq 0$ and so,

$$\begin{aligned} \sum_{v \in V} \left(\mathbb{P}(\hat{Y}_v \neq Z_v) - \mathbb{P}(Y_v^* \neq Z_v) \right) &\leq \left(\frac{2}{3} + \frac{c}{2} \right) \log(|\mathcal{F}|/\delta) + \frac{1}{c} \sum_{v \in V} \mathbb{E} \left[\left(\mathbb{1}\{\hat{Y}_v \neq Z_v\} - \mathbb{1}\{Y_v^* \neq Z_v\} \right)^2 \right] \\ &= \left(\frac{2}{3} + \frac{c}{2} \right) \log(|\mathcal{F}|/\delta) + \frac{1}{c} \sum_{v \in V} \mathbb{1}\{\hat{Y}_v \neq Y_v^*\}. \end{aligned}$$

□

Lemma 6 (Maximal Inequality). For each $i \in [N]$, let $\{T_v^i\}_{v \in V}$ be a random process with each variable T_v^i bounded in absolute value by 1. Define $\sigma_i^2 = \sum_{v \in V} \text{Var}(T_v^i)$. With probability at least $1 - \delta$,

$$\sum_{v \in V} (\mathbb{E}[T_v^i] - T_v^i) \leq \frac{2}{3} \log(N/\delta) + \sqrt{2\sigma_i^2 \log(N/\delta)} \quad \forall i \in [N]. \quad (27)$$

Proof of Lemma 6. Let us start by writing out the Bernstein bound for the random variable $\sum_{t=1}^n Z_t^i$:

$$\mathbb{P} \left(\sum_{v \in V} (\mathbb{E}[T_v^i] - T_v^i) > \theta \right) \leq \exp \left(-\frac{\theta^2}{2\sigma_i^2 + \frac{2}{3}\theta_i} \right).$$

We now consider the family of processes $\{T_v^i\}_{v \in V}$ and see that by union bound we have

$$\mathbb{P} \left(\max_{i \in [N]} \sum_{v \in V} (\mathbb{E}[T_v^i] - T_v^i) - \theta_i > 0 \right) \leq \sum_{i \in [N]} \exp \left(-\frac{\theta_i^2}{2\sigma_i^2 + \frac{2}{3}\theta_i} \right).$$

Solving the quadratic formula we see that if we take

$$\theta_i \geq \frac{1}{3} \log(N/\delta) + \sqrt{\log^2(N/\delta)/9 + 2\sigma_i^2 \log(N/\delta)},$$

then we have

$$\sum_{i \in [N]} \exp \left(-\frac{\theta_i^2}{2\sigma_i^2 + \frac{4}{3}} \right) \leq \delta.$$

We can conclude that

$$\mathbb{P} \left(\forall i \in [N], \sum_{v \in V} (\mathbb{E}[T_v^i] - T_v^i) > \frac{1}{3} \log(N/\delta) + \sqrt{\log^2(N/\delta)/9 + 2\sigma_i^2 \log(N/\delta)} \right) \leq \delta.$$

□

D Algorithms

The tree inference algorithm from [Section 2](#) and the full tree decomposition inference algorithm, [Algorithm 1](#), rely on the solution of a constrained minimization problem over the edges and vertices of a tree. This minimization problem is stated in its most general form as [Algorithm 2](#). This problem can be solved efficiently using the following tree-structured graphical model:

- Fix an arbitrary order on T , and let $p(v)$ denote the parent of a vertex v under this order.
- Define variables $s \in \{\pm 1\}^V$ and $C \in \{1, \dots, K_n\}^V$.
- For each variable $v \in V$ define factor:

$$\psi_v(s_v, s_{p(v)}, C_v, C_{\delta_+(v)}) = e^{-\mathbb{1}\{\text{Cost}_v[s_v]\}} \cdot \mathbb{1} \left\{ \sum_{u \in \delta_+(v)} C_u \leq C_v - \mathbb{1}\{s_v \neq s_{p(v)} \cdot S(v, p(v))\} \right\}.$$

With this formulation it is clear that given (s, C) maximizing the potential

$$\psi(s, C) = \prod_{v \in V} \psi_v(s_v, s_{p(v)}, C_v, C_{\delta_+(v)})$$

the node labels s are a valid solution for [Algorithm 2](#). Since ψ is a tree-structured MRF the maximizer can be calculated exactly using max-sum message passing (see e.g. [Cowell et al. \(2006\)](#)). The only catch is that naively this procedure's running time will scale as $n^{\deg(T)}$, because each of the variables C_v has a range that scales with n . For example, the range of C_v is $\tilde{O}(pn)$ for the setup in [Section 2](#). We now show that the structure of the factors can be exploited to perform message passing in polynomial time in $\deg(T)$ and n . In particular, message passing can be performed in time $\tilde{O}(\deg(T)K_n n^3)$ for general trees and time $\tilde{O}(K_n n^2)$ when T is a line graph.

Algorithm 2 TREEDECODER

Input: Tree $T = (V, E)$, $\{\text{Cost}_v\}_{v \in V}$, $\{S(u, v)\}_{(u, v) \in E}$, $K_n \in \mathbb{N}$.

$$\hat{s} = \arg \min_{s \in \{\pm 1\}^V} \sum_{v \in V} \text{Cost}_v[s_v]$$

$$\text{s.t.} \quad \sum_{(u, v) \in E} \mathbb{1}\{s_u \neq s_v \cdot S(u, v)\} \leq K_n$$

Return: $\hat{s} \in \{\pm 1\}^V$.

To solve TREEDECODER efficiently, we first turn T into a DAG by running a BFS from a given vertex r and directing edges according to the time of discovery. We denote this DAG by \vec{T} . We root this directed tree at r , and denote the parent of a vertex $u \neq r$ by $p(u)$. For $u \in V$, let \vec{T}_u denote the (directed) subtree rooted at u . Given a labeling Y to the vertices of T , an edge uv for which $s_u \neq s_v \cdot S(u, v)$ is called a *violated edge*.

We now define a table OPT that will be used to store values for sub-problems of Algorithm 2. For $u \neq r$, and budget K , we define $OPT(u, K|1)$ to be the optimal value of the optimization problem in Algorithm 2 over the subtree \vec{T}_u for budget K , where the label of $p(u)$ is constrained to have value 1. Importantly, the edge $(u, p(u))$ is also considered in the count of violated edges (in addition to the edges in \vec{T}_u). $OPT(u, K|-1)$ is defined likewise, but for $p(u)$ constrained to label value -1 .

$$OPT(u, K|1) = \min_{s \in \{-1, 1\}} \min_{\sum_{v \in N_u} K_v = K - \mathbb{1}_{\{s \neq S_{p(u)} \cdot S(u, p(u))\}}} \left(\sum_{v \in N(u)} OPT(v, K_v|s) + \text{Cost}_v[s] \right).$$

Here s is simply the value assigned to u . We constrain the budgets K_v to satisfy $0 \leq K_v \leq |\vec{T}_v|$ (clearly no subtree \vec{T}_v can violate more than $|\vec{T}_v|$ edges). For the sake of readability, we do not include this constraint in the recursive formula above. A similar recursion can be obtained for $OPT(u, K|-1)$.

One can verify that if we can compute $OPT(u, K|s)$ for all nonroot nodes and all values of $K \leq K_n, s \in \{-1, 1\}$ then we can find the optimum of the problem of our whole tree. To achieve this, simply attach a degree one node r' to the root of the tree, add a directed edge (r', r) and set the label of the root to equal 1. Then we simply solve for $OPT(r, K|1)$, where $S(r, r') = 1$ as well as $OPT(r, K, 1)$, where $S(r', r)$ is -1 and return the minimum of the the values.

For a leaf node w , the value of $OPT(w, K'|s)$ can be calculated as follows: it is $\min(\text{cost}[s_w = -1], \text{cost}[s_w = 1])$, for $K' \geq 1$. If $K = 0$, it is $\text{cost}[s']$ where s' is the unique label not violating the constraint $s \neq s' \cdot S(w, p(w))$.

We now show how to calculate $OPT(u, K_u|s)$ for any vertex in the tree, assuming OPT has already been calculated for its children. To do this, we try both values of s_u , and then condition on its value to optimize

$$\min_{\sum_{j \in [1, k]} K_j = K - \mathbb{1}_{\{s \neq s_{p(u)} \cdot S(u, p(u))\}}} \sum_{u \in [1, k]} OPT(j, K_j|s).$$

The function $\sum_{v \in N_u} OPT(v, K_v|s)$ can be minimized using another layer of dynamic programming as follows: For $r \leq s$, let $[r, s]$ be the set of integers between r and s . Assuming we enumerate the vertices in $N(u)$ by $1, \dots, k := |N(u)|$ and setting K_j to be the budget for the j th node, we have the equality

$$\begin{aligned} & \min_{\sum_{j \in [1, k]} K_j = K - \mathbb{1}_{\{s \neq s_{p(u)} \cdot S(u, p(u))\}}} \sum_{u \in [1, k]} OPT(j, K_j|s) \\ &= \min_{K_1 \in [0, K - \mathbb{1}_{\{s \neq s_{p(u)} \cdot S(u, p(u))\}}]} OPT(1, K_1|s) + \min_{\sum_{j \in [2, k]} K_j = K - K_1 - \mathbb{1}_{\{s \neq s_{p(u)} \cdot S(u, p(u))\}}} \sum_{j \in [2, k]} OPT(j, K_j|s). \end{aligned}$$

The minimization problem can be solved in time $O(|N(u)|K_n^2)$ time. We first calculate the minimum cost for the first two vertices where the number of constraints violated can range between 1 to K . This can be done in time $O(K^2)$. We then examine the minimum cost for the first

three vertices (assuming of course u has at least three descendants) where the number of violated constraints ranges between 0 and K . Since we have the information for the first two vertices, these values can be calculated again in time $O(K^2)$. We repeat this iteration until all descendants of u are considered. It follows that the overall running time of this algorithm is $\sum_{u \in V} |N(u)|K_n^2 = O(nK_n^2)$, since T is a tree.

When T is a line graph each node has a single child, the recursion collapses to time $O(nK_n)$.

E Further Techniques for General Graphs

Here we give a simple proof that if the minimal degree of G is $\Omega(\log n)$, then there is an algorithm that achieves arbitrarily small error for each vertex as $n \rightarrow \infty$ as soon as $q = 1/2 - \epsilon$ is constant.

Theorem 6. *There is an efficient algorithm that guarantees*

$$\mathbb{E} \left[\sum_{v \in V} \mathbb{1} \{ \hat{Y}_v \neq Y_v \} \right] \leq \sum_{v \in V} \exp(-C \deg(v) \epsilon^2 (1 - 2p)^2).$$

for some $C > 0$.

Observe that this rate quickly approaches 0 with n as soon as $\deg(G) = \Omega(\log n)$ (i.e., it has $o(n)$ Hamming error). On the other hand, if degree is constant (say d), then even when $p = 0$ the rate of this algorithm is only $e^{-dO(\epsilon^2)}n$, so the algorithm does not have the desired property of having error approach 0 as $p \rightarrow 0$.

Proof. Fix a vertex v and, for each vertex u in its neighborhood, define an estimate $S_u = Z_u \cdot X_{uv}$. We can observe that $\mathbb{P}(S_u = Y_v) = (1 - p)(1 - q) + pq = \frac{1}{2} + \epsilon(1 - 2p)$. Our algorithm will be to use the estimator $\hat{Y}_v = \text{Majority}(\{S_u\}_{u \in N(v)})$. Since each S_u is independent, the Hoeffding bound gives that

$$\mathbb{P}(\hat{Y}_v \neq Y_v) \leq \exp(-C \deg(v) \epsilon^2 (1 - 2p)^2).$$

Taking this prediction for each vertex gives an expected hamming error bound of

$$\mathbb{E} \left[\sum_{v \in V} \mathbb{1} \{ \hat{Y}_v \neq Y_v \} \right] \leq \sum_{v \in V} \exp(-C \deg(v) \epsilon^2 (1 - 2p)^2).$$

□